

# Method for hue plane preserving color correction

MICHAL MACKIEWICZ,<sup>1,\*</sup> CASPER F. ANDERSEN,<sup>2</sup> AND GRAHAM FINLAYSON<sup>1</sup>

<sup>1</sup>School of Computing Sciences, University of East Anglia, Norwich, UK

<sup>2</sup>The Norwegian Color Research Laboratory, Gjøvik University College, Gjøvik, Norway

\*Corresponding author: M.Mackiewicz@uea.ac.uk

Received 22 April 2016; revised 11 September 2016; accepted 12 September 2016; posted 12 September 2016 (Doc. ID 261628); published 14 October 2016

Hue plane preserving color correction (HPPCC), introduced by Andersen and Hardeberg [Proceedings of the 13th Color and Imaging Conference (CIC) (2005), pp. 141–146], maps device-dependent color values ( $RGB$ ) to colorimetric color values ( $XYZ$ ) using a set of linear transforms, realized by white point preserving  $3 \times 3$  matrices, where each transform is learned and applied in a subregion of color space, defined by two adjacent hue planes. The hue plane delimited subregions of camera  $RGB$  values are mapped to corresponding hue plane delimited subregions of estimated colorimetric  $XYZ$  values. Hue planes are geometrical half-planes, where each is defined by the neutral axis and a chromatic color in a linear color space. The key advantage of the HPPCC method is that, while offering an estimation accuracy of higher order methods, it maintains the linear colorimetric relations of colors in hue planes. As a significant result, it therefore also renders the colorimetric estimates invariant to exposure and shading of object reflection. In this paper, we present a new flexible and robust version of HPPCC using constrained least squares in the optimization, where the subregions can be chosen freely in number and position in order to optimize the results while constraining transform continuity at the subregion boundaries. The method is compared to a selection of other state-of-the-art characterization methods, and the results show that it outperforms the original HPPCC method.

Published by The Optical Society under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

**OCIS codes:** (330.1690) Color; (130.6010) Sensors.

<http://dx.doi.org/10.1364/JOSAA.33.002166>

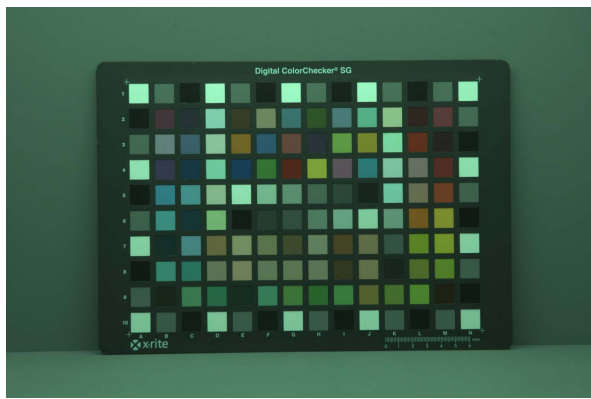
## 1. INTRODUCTION

In digital photography, camera characterization relates the camera output values, known as device-dependent  $RGB$ , to standard observer colorimetric values, known as device-independent  $XYZ$  [1] (or to a known linear combination of  $XYZ$ , such as  $sRGB$  [2]). However, mapping  $RGB$  to  $XYZ$  cannot be precisely achieved. The reason for this is that cameras do not sample light in the same way as  $XYZ$ . Mathematically, camera sensors do not span the same subspace as the human visual system, i.e., camera sensors are not linear transforms from the  $XYZ$  color matching functions [3,4] (we say that they are not colorimetric). This is mainly due to noise considerations [5] and limitations in the manufacturing process. Given that the camera sees, or samples, light differently than the human visual system, color correction is a pragmatic step for best mapping  $RGBs$  to  $XYZs$ . The simplest and, it turns out, the most widely used color correction strategy is to map  $RGBs$  to  $XYZs$  using a  $3 \times 3$  color correction matrix. The above method is called the linear color correction (LCC) [6]. In Fig. 1, we can see an input image as captured by the NIKON D70 camera,

which we call a raw image, and an image corrected to the  $sRGB$  color space using the LCC.

In Fig. 2, three sets of spectral sensor functions are shown. In Fig. 2(b), the color matching functions of the human standard observer ( $XYZ$ ) [1] and in Fig. 2(a) a set of digital camera sensors [7] are plotted as functions of wavelength. It is readily seen that the two sets of sensors are different and the responses from them therefore also must be different. The best linear transform of the camera sensors to the  $XYZ$  color matching functions with respect to the sensor RMSE is shown in Fig. 2(c).

In order to improve on the performance of the LCC, numerous more advanced methods have been proposed. They include polynomial regression, nonlinear optimization, neural networks, and look-up tables (LUTs). For the conversion of color signals from digital color scanning, which is strongly related to digital photography, polynomial regression of different polynomial degrees was investigated in Ref. [8] and a spectral scanner model was used in Ref. [9]. In Ref. [10], polynomial modeling was tested on digital camera characterization. Neural networks are used in Refs. [11–13] and compared to polynomial



(a) Raw image.



(b) Corrected to sRGB.

**Fig. 1.** Nikon D70 raw camera response to the scene containing a color checker, before (a) and after (b) correction to the *sRGB* color space by means of a  $3 \times 3$  color correction matrix. Both images have a gamma of 0.5 applied.

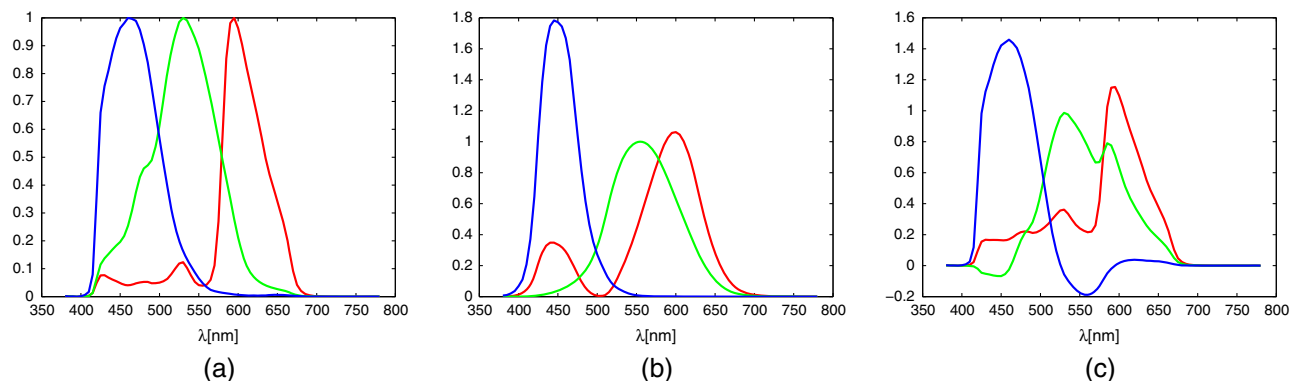
methods in Ref. [14]. Methods of device calibration using look-up tables and interpolation are given in Ref. [15], and a hybrid of look-up table and linear transformation is presented in Ref. [16]. In order to remedy noise effects in the color correction, a method of varying linear color transformation as a function of noise estimates in spatially local areas of a color image is presented in Ref. [17]. Color correction can be based on optimization of colorimetric differences, which is most common, or on optimization of nonlinear perceptual

differences. The latter include a color correction method based on altering the LCC transform coefficients using a spherical sampling approach [18].

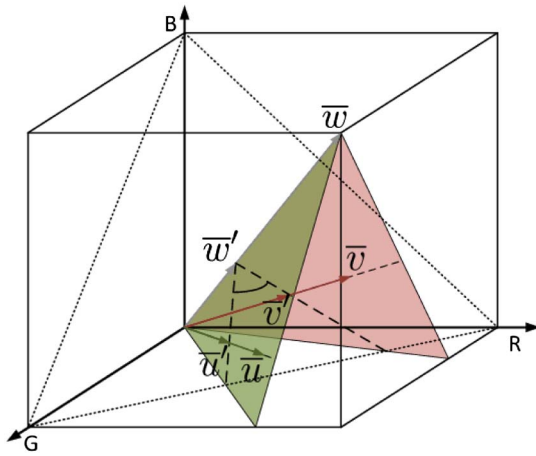
A method introducing constrained linear color correction was presented in Ref. [19] to enforce a mapping of one (or possibly two) color(s) without error. The authors called their method the white preserving color correction (WPCC) as the natural choice of the color to be mapped exactly is neutral. Perceptually, it is typically very important that scene neutral and white are mapped to colorimetric neutral and white precisely in order to avoid color casts in the rendered image [20]. The method can be also applied within the framework of the polynomial color correction.

Exposure and shading independence by which scalings of reflectances recorded by the camera device are mapped to the exact same scalings of the reflectances matched by the observer is an inevitable priority when colorimetric correctness is desired. If the device response to the reflectance of a surface is recorded with values  $C_{rgb}$  and correspondingly  $C_{xyz}$  by the observer, then these responses would yield  $kC_{rgb}$  and  $kC_{xyz}$  if exposure time or irradiance on the surface was modified by a scalar value of  $k$ . In practice, this scaling takes place when surfaces locally occur in a scene with different irradiances or globally as a function of exposure time. The  $3 \times 3$  matrix method is inherently exposure and shading independent and will therefore map these scalings correctly. The root-polynomial color correction [21] is also exposure independent.

The precursor to our method is the HPPCC method [22]. Here, linear transforms are applied per hue segment in such a way that color correction is hue preserving, white point preserving, and continuous. The method operates on subregions of color space, defined by two adjacent hue planes. These hue plane delimited subregions of white-balanced camera *RGB* values are mapped to corresponding hue plane delimited subregions of estimated colorimetric *XYZ* values. Hue planes are geometrical half-planes, each defined by the neutral axis and a chromatic color in a linear color space. On an intersecting plane perpendicular to the neutral axis, an angle, known as a hue angle of the hue plane, can be measured. Between two such adjacent hue planes, a convex cone in the color space is defined. The cone has its vertex in the origin and it contains the full gamut of colors between the two hue planes that were recorded by the camera sensors or the color matching functions (see Fig. 3).



**Fig. 2.** (a) Example of real camera (Nikon 5100) sensors [7]; (b) *XYZ* color matching functions; and (c) their estimate using the best linear transform (least squares).

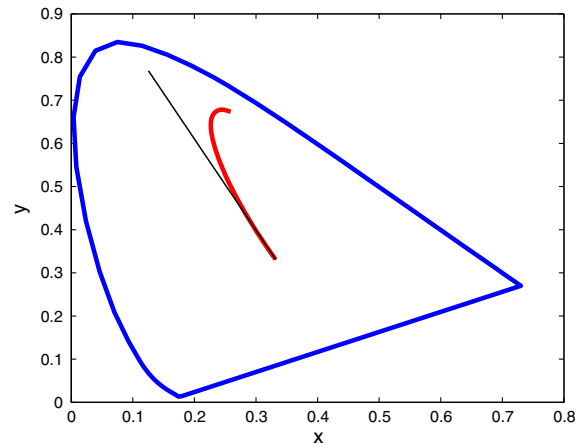


**Fig. 3.** Visualization of the convex cone spanned by the two hue planes in the  $RGB$  unit cube. A hue plane is spanned by the neutral vector  $\bar{w}$  and a chromatic color ( $\bar{u}$  or  $\bar{v}$ ). These three vectors intersect the chromaticity plane (dotted triangle) at  $\bar{w}'$ ,  $\bar{u}'$ , and  $\bar{v}'$ . The hue planes intersect the chromaticity plane in hue lines (dashed lines).

Hue plane preservation enables an exact mapping of linear relations of colors that originate from a linear combination of a neutral reflection or a specular highlight and a diffuse reflection. If, for example, an object or a surface has a reflection that can be entirely (or at least approximately) characterized by a mixture of a neutral reflection with the color of the illuminant and a reflection of a chromatic surface (i.e., diffuse reflection) [23], then the device response can be written as  $C_{rgb}(k_w, k_d) = k_w W_c + k_d D_c$ . The corresponding response from the observer is  $C_{xyz}(k_w, k_d) = k_w W_o + k_d D_o$ , where  $W_c$  and  $W_o$  are the responses to the neutral,  $D_c$  and  $D_o$  are the responses to the chromatic surface, and  $k_w$  and  $k_d$  are the scalars that determine the mixture. By varying  $k_w$  and  $k_d$ , hue planes are described in device and observer space. In order for a mapping  $f()$  to be hue plane preserving, the equation  $f(C_{rgb}(k_w, k_d)) = C_{xyz}(k_w, k_d)$  must hold for all  $k_w$  and  $k_d$ .

It can be easily seen that the LCC is hue plane preserving. However, the polynomial and root-polynomial mappings will not be hue plane preserving, despite their potential higher accuracy. They will distort the hue planes, which can lead to undesirable perceptual deviations of color, i.e., hue shifts and possibly unstable predictions of very saturated colors (see Fig. 4).

The major contributions of this work can be described as follows. The original HPPCC method required a careful selection of the small set of color samples that would both delimit the color space subregions and define the linear color correction transforms within them. Here, we pose a color correction problem as the colorimetric error minimization of all available color samples while preserving the geometrical structure of separate and continuous linear transforms as in the original HPPCC. Furthermore, our framework allows for more flexible optimization with respect to the location of the color space subregion boundaries. In Section 2 we give details of the original and in Section 3 the proposed algorithm. The method proposed in this



**Fig. 4.** CIE  $xy$  chromaticity diagram and a sample hue plane distortion resulting from the root-polynomial color correction of degree two (red [21]) and nonmodified hue line (black).

paper significantly improves color correction results comparing with its precursor, which is shown in Section 4.

The preliminary version of this algorithm has been presented at a conference [24]. Here, we present an extended version of this work, which includes an additional step enabling optimization of the hue region boundaries, and give additional results and discussions. Comparing to the conference publication, the final method utilizing the aforementioned optimization offers modest improvements of the color correction for the algorithm settings with a low number of hue regions, i.e., for simpler models that require less parameters.

## 2. HUE PLANE PRESERVING COLOR CORRECTION

In this section, we will briefly describe the original hue plane preserving color correction method.

Throughout this paper, we will always assume that we deal with the  $RGB$  color responses that were white-balanced; that is, each  $RGB$  response has been divided by the respective color response of the perfect diffuser denoted as  $\bar{w} = (R_N, G_N, B_N)$ .

Andersen and Hardeberg [22] proposed to sort the  $RGB$  color responses according to their hue correlates  $\theta_i$  in the  $rg$  chromaticity plane in ascending order as

$$\theta_i = \begin{cases} 0 & \text{if } r_i = g_i = \frac{1}{3}, \\ \frac{\pi}{2} & \text{if } r_i = \frac{1}{3} \wedge g_i > \frac{1}{3}, \\ \frac{3\pi}{2} & \text{if } r_i = \frac{1}{3} \wedge g_i < \frac{1}{3}, \\ \arctan \frac{g_i - \frac{1}{3}}{r_i - \frac{1}{3}} + m\pi & \text{otherwise,} \end{cases} \quad (1)$$

where

$$m = \begin{cases} 0 & \text{if } r_i \geq \frac{1}{3} \wedge g_i \geq \frac{1}{3}, \\ 1 & \text{if } g_i < \frac{1}{3}, \\ 2 & \text{otherwise,} \end{cases} \quad (2)$$

and the chromaticity values are calculated by

$$r_i = R_i / (R_i + G_i + B_i), \quad g_i = G_i / (R_i + G_i + B_i). \quad (3)$$

Next, they choose  $K$  chromatic  $RGB$  samples denoted as  $\bar{q}_k$  that will define the boundaries between  $K$  hue regions. Finally, they find the  $3 \times 3$  color correction transform for each region  $k$  by inverting the following equation:

$$\begin{aligned} \begin{bmatrix} \bar{q}_k^T \\ \bar{q}_{k+1}^T \\ \bar{w}^T \end{bmatrix} \mathbf{T}_k &= \begin{bmatrix} \bar{p}_k^T \\ \bar{p}_{k+1}^T \\ \bar{p}_w^T \end{bmatrix} && \text{for } k = 1 \dots k-1 \\ \begin{bmatrix} \bar{q}_k^T \\ \bar{q}_1^T \\ \bar{w}^T \end{bmatrix} \mathbf{T}_k &= \begin{bmatrix} \bar{p}_k^T \\ \bar{p}_1^T \\ \bar{p}_w^T \end{bmatrix} && \text{for } k = k, \end{aligned} \quad (4)$$

where  $\bar{p}_k$  denotes a measured  $XYZ$  corresponding to the color response  $\bar{q}_k$ , and  $\bar{p}_w$  denotes a measured  $XYZ$  corresponding to the color response of the neutral patch  $\bar{w}$ . By construction,  $\mathbf{T}_k$  is invertible and hence all color samples at the boundaries of the hue regions as well as the neutral sample are mapped exactly to their corresponding  $XYZ$  values. The above also ensures the continuity of the transform. The second line in Eq. (4) guarantees the transform wraparound with respect to hue, i.e., the last hue region extends from  $\theta_K$  to  $\theta_1$ . The construction of the hue regions in the  $rg$  chromaticity space is illustrated in Fig. 5. The figure shows four hue regions extending from the white point  $\bar{w}$  whose extent is limited by the two consecutive hue region boundaries.

Thus, in a preprocessing step, the HPPCC method calculates  $K$ ,  $3 \times 3$  matrices—one for each hue region. When the color correction is applied to the image, they map each  $RGB$  pixel into its chromaticity and calculate its hue angle using Eqs. (1)–(3). Then, they use the hue region boundary samples  $\bar{q}_k$  to calculate the hue region, with respect to which a pixel belongs. Finally, they apply the color correction matrix corresponding to this hue region.

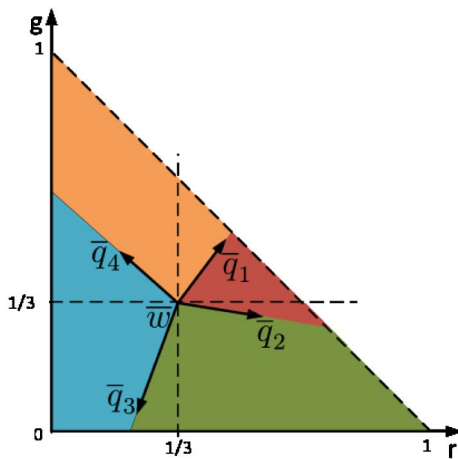


Fig. 5. Construction of hue regions in the  $rg$  chromaticity space.

### 3. HUE PLANE PRESERVING COLOR CORRECTION USING CONSTRAINED LEAST SQUARES

Here, we present in detail our new hue plane preserving color correction using constrained least squares.

In the HPPCC method, the matrices are found given a set of known  $RGB$ s and known corresponding  $XYZ$ s that divide the color space into the number of hue regions. However, in this paper, we wish to find matrices that “best fit” all our data.

We denote an  $N \times 3$  matrix of camera responses as  $\mathbf{Q}$  and an  $N$ -vector of corresponding measured  $X$  responses as  $\bar{x}$ . The linear least-squares color correction is given as

$$\text{minimize}_{\bar{t}} \|\mathbf{Q}\bar{t} - \bar{x}\|, \quad (5)$$

where  $\bar{t}$  is a 3-vector to be found. To be precise, in color correction, we must also solve for the mapping to the other two channels, i.e.,  $Y$  and  $Z$ , but the math is the same, so below we will only solve for  $RGB$  to  $X$ .

In the method proposed in this paper, we alter the above optimization by sorting  $N$  color responses according to  $\theta$  and dividing them into  $K$  subsets. Each subset contains  $N_k$  samples such that  $\sum_{k=1}^K N_k = N$ . The subset samples are placed into the rows of  $K$ ,  $N_k \times 3$  matrices denoted as  $\mathbf{Q}_k$ . Next,  $\bar{x}$  is sorted according to the same index and partitioned into  $K$  subsets denoted as  $\bar{x}_k$ . We can seek for a separate solution to color correction for each subset  $k$  of the dataset. This can be achieved by performing the following optimization:

$$\text{minimize}_{\bar{t}_1 \dots \bar{t}_K} \sum_{k=1}^K \|\mathbf{Q}_k \bar{t}_k - \bar{x}_k\|. \quad (6)$$

Note that the above returns different correction transforms  $\bar{t}_k$  for each subset of the training set and the solution for  $\bar{t}_k$  is independent of  $\bar{t}_j$ . HPPCC has the important property that the color correction transform is continuous, and this is because the two colors defining the hue boundary (the neutral and an  $RGB$ ) are the same for two successive color correction transforms. Solved independently, Eq. (6) would not have this property. Thus, we are going to pose the least-squares problem as if the solutions were coupled so we can solve for all the unknown variables at the same time.

Let us rewrite Eq. (6) as

$$\text{minimize}_{\bar{T}} \|\mathbf{A}\bar{T} - \bar{X}\|, \quad (7)$$

where  $\mathbf{A}$  is a nonsquare ( $N \times 3K$ ) block diagonal matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q}_1 & 0 & \dots & 0 \\ 0 & \mathbf{Q}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{Q}_K \end{bmatrix}, \quad (8)$$

$\bar{T}$  is a  $3K$ -vector  $\bar{T} = (\bar{t}_1^T, \dots, \bar{t}_K^T)^T$ , and  $\bar{X}$  is an  $N$ -vector  $\bar{X} = (\bar{x}_1^T, \dots, \bar{x}_K^T)^T$ .

Let us denote an  $RGB$  vector whose corresponding hue angle is greater than the hue angle of the last sample in  $\mathbf{Q}_k$  and less than the hue angle of the first sample in  $\mathbf{Q}_{k+1}$  as  $\bar{q}_k^T$ . This definition includes a “wraparound,” i.e., an  $RGB$  separating the last (that is the  $K$ th) hue region from the first needs to have

its hue angle between the hue angles of the last sample in  $\mathbf{Q}_K$  and the first sample in  $\mathbf{Q}_1$ .

Then we can formulate the following equality constraints:

$$\begin{aligned}\bar{q}_k^T \bar{t}_k &= \bar{q}_k^T \bar{t}_{k+1}, & k = 1, \dots, K-1, \\ \bar{q}_k^T \bar{t}_k &= \bar{q}_k^T \bar{t}_1, & k = K, \\ \bar{w}^T \bar{t}_1 &= \bar{w}^T \bar{t}_2 = \dots = \bar{w}^T \bar{t}_K = x_w,\end{aligned}\quad (9)$$

where  $\bar{w}$  and  $x_w$  are a white-balanced camera response and a colorimetric response  $X$  to the neutral patch, respectively. In total, there are  $2K$  constraints.

These constraints can be written in a matrix form as

$$\mathbf{C}\bar{T} = \bar{b}, \quad (10)$$

where  $\mathbf{C}$  is a  $2K \times 3K$  matrix:

$$\mathbf{C} = \begin{bmatrix} \bar{q}_1^T & -\bar{q}_1^T & 0 & \dots & 0 \\ 0 & \bar{q}_2^T & -\bar{q}_2^T & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \bar{q}_{K-1}^T & -\bar{q}_{K-1}^T \\ -\bar{q}_K^T & 0 & \dots & 0 & \bar{q}_K^T \\ \bar{w}^T & -\bar{w}^T & 0 & \dots & 0 \\ 0 & \bar{w}^T & -\bar{w}^T & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \bar{w}^T & -\bar{w}^T \\ \bar{w}^T & \dots & 0 & \dots & 0 \end{bmatrix}, \quad (11)$$

and

$$\bar{b} = \begin{bmatrix} \bar{0} \\ x_w \end{bmatrix}. \quad (12)$$

The new optimization is

$$\begin{aligned} & \text{minimize } \|\mathbf{A}\bar{T} - \bar{X}\| \\ & \text{subject to } \mathbf{C}\bar{T} = \bar{b}. \end{aligned}\quad (13)$$

This constrained optimization enforces continuity of the color correction transform at the boundaries of the hue regions. Further, it can be noted, that if one wishes to preserve the continuity of color correction transform, it is not necessary to have the last constraint that enforces white preservation [the last equality in the last row in Eq. (9) and consequently the last rows in Eqs. (11) and (12)]. If we omitted this constraint, then  $\mathbf{C}$  would be a  $(2K-1) \times 3K$  matrix and  $\bar{b}$  would be a zero vector. While the reduced set of constraints could provide a good solution to the color correction problem, we chose to supplement it with an additional white preserving constraint. This is in line with both the original HPPCC method as well as the white preserving color correction [19] that was mentioned in Section 1.

The above optimization can be written for all three  $XYZ$  channels as

$$\begin{aligned} & \text{minimize } \|\mathbf{A}\mathbf{T} - \mathbf{X}\| \\ & \text{subject to } \mathbf{C}\mathbf{T} = \mathbf{B}, \end{aligned}\quad (14)$$

where  $\mathbf{X}$  is an  $N \times 3$  matrix of measured and sorted  $XYZs$ ,  $\mathbf{T}$  is a  $3K \times 3$  matrix consisting of  $K$ ,  $3 \times 3$  linear color correction matrices, and

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \bar{p}_w^T \end{bmatrix}, \quad (15)$$

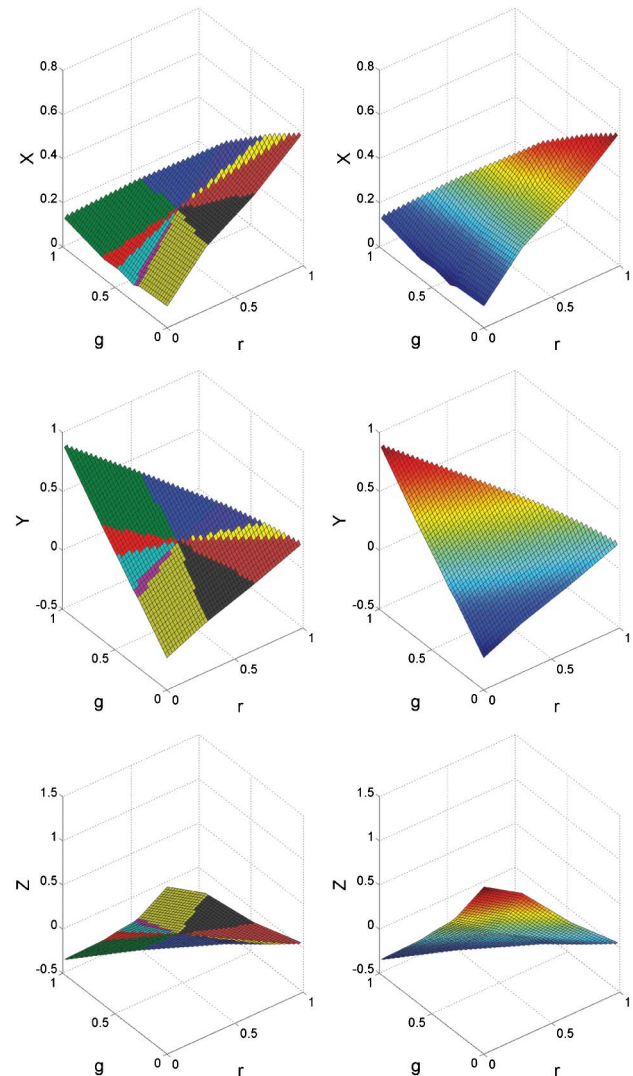
where  $\bar{p}_w^T$  is a row vector of measured  $XYZs$  of the neutral patch, and  $\mathbf{0}$  a  $(2K-1) \times 3$  matrix containing zeros.

This optimization can be solved using the method of Lagrange multipliers, which provides the following closed-form solution [25,26]:

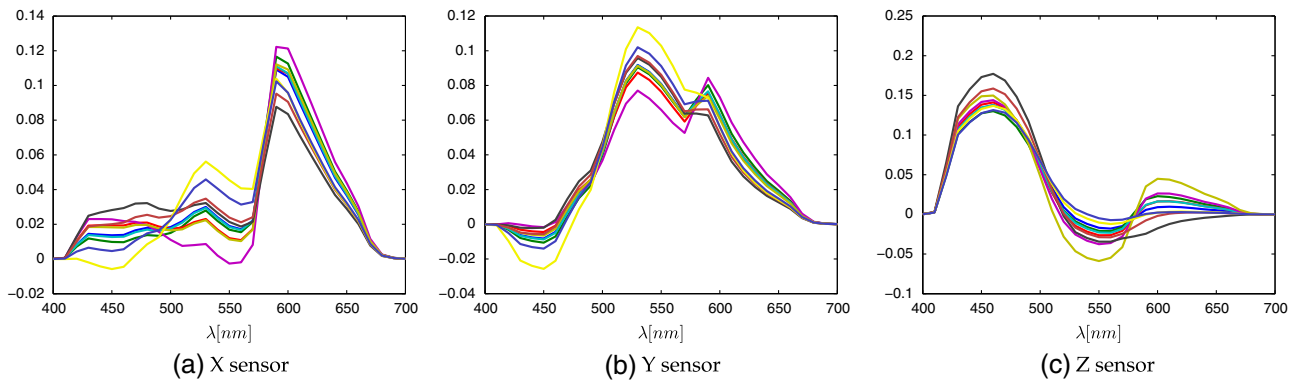
$$\begin{bmatrix} \mathbf{T} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} 2\mathbf{A}^T\mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} 2\mathbf{A}^T\mathbf{X} \\ \mathbf{B} \end{bmatrix}, \quad (16)$$

where  $\mathbf{Z}$  is a matrix of Lagrange multipliers.

In Fig. 6, we can see a visualization of the color correction transformation calculated using the method described above. It



**Fig. 6.** Cross sections along the  $rg$  chromaticity plane of the  $XYZ = f(RGB)$  color correction hypersurface. Left: different colors representing ten hue regions used in this example. Right: the same figure with coloring proportional to  $X$  (top),  $Y$  (middle), and  $Z$  (bottom).



**Fig. 7.** Estimated ten sets of color matching functions. The colors of the plots correspond to the colors of the hue regions in Fig. 6.

can be seen that the surface created from the transformed chromaticities is continuous and consists of a number of planes anchored at the white point. Clearly, the calculated transforms are different in different hue regions. We can get a further intuition of how different the color transforms are in different hue regions if we compare the *RGB* camera sensor sensitivities transformed by the color correction transforms optimized for those different hue regions. In this example, the color correction transform was calculated for the dataset generated using Nikon camera spectral sensitivities [see Fig. 2(a)]. In Fig. 7, we can see the estimated *XYZ* color matching functions resulting from the color correction transformation applied to the above *RGB* sensors. We can see that the estimated *XYZ* CMFs are clearly different in different hue regions. In particular, the ten *Y* estimates differ among each other perhaps as much as the *X* and *Z* estimates, which is not immediately apparent when comparing the rows of Fig. 6.

It is interesting to note the number of degrees of freedom (DFs) that can be optimized in our proposed framework. Matrix  $\mathbf{C}$  is (in general) rank  $2K$ . Then, from the Rank-nullity theorem, we have that  $\text{nul}(\mathbf{C}) = K$ , which is a number of DFs to be optimized in the case of a single channel color correction. Thus, for  $K = 3$ , we will have the same number of degrees of freedom to be optimized as in the linear color correction—3 for single channel color correction or  $3 \times 3 = 9$  in the three channel case. Analogously, for  $K = 4$ , we have  $4 \times 3 = 12$  DFs, for  $K = 5$ , we have  $5 \times 3 = 15$  DFs, etc. The situation is somewhat different in the  $K = 2$  case. Then, the color transforms in the two hue regions are different only when  $\tilde{q}_1$ ,  $\tilde{q}_2$ , and  $\tilde{w}$  are coplanar (equivalent to  $\theta_2 = \theta_1 + \pi$ ), in which case  $\text{rk}(\mathbf{C}) = 3$  and as in the  $K = 3$  case, we have  $3 \times 3 = 9$  DFs.

### A. Hue Region Boundary Optimization

Above, we described the optimization method for calculating a color correction transform subject to white preserving and continuity constraints at the boundaries of some given hue regions. Here, we are considering the possible ways for finding the optimal locations in the hue circle of the hue boundaries  $\theta_k$  and their corresponding  $\tilde{q}_k$  *RGBs* that are used to form the constraints.

Splitting the set of color samples into the  $K$  subsets corresponding to the  $K$  different hue regions with an equal number of samples in each hue region would be one way of doing this.

While this idea may provide good results, it is very unlikely that such an approach will give an optimal performance from the point of view of color correction accuracy, in particular for a low number of hue regions. Hence, we think that there is certainly some scope for further optimization of the hue region boundaries.

The optimization that we have chosen utilizes gradient descent. The initial parameters (hue region boundaries) are calculated as described in the above paragraph, i.e., initially hue regions contain the same number of color samples. Then, the parameters of the color correction transform  $\mathbf{T}$  and the hue region boundaries  $\theta_k$  are updated iteratively. In each iteration, the parameters of the color correction transform  $\mathbf{T}$  are recalculated for the updated set of  $\theta_k$  and the training set mean error is calculated in the CIELUV color space. A gradient descent algorithm is used to minimize this error, which returns a set of  $\theta_k$  (and  $\mathbf{T}$ ) parameters corresponding to a (local) error minimum.

The above optimization procedure is performed if the number of required hue regions is greater than two. If this is not the case, i.e., we require only two hue regions, then we must employ a slightly different approach. The reason for this is that if we want the two color transforms to be different, then as mentioned in Section 3, we must ensure that  $\tilde{q}_1$ ,  $\tilde{q}_2$ , and  $\tilde{w}$  are coplanar, which is equivalent to  $\theta_2 = \theta_1 + \pi$ . Hence, in this case we are searching for one parameter:  $\theta_1$ . This can be achieved by performing an exhaustive search giving us an initial  $\theta_1$  value, followed by further gradient descent optimization.

For a high number of hue regions, the gradient descent optimization may produce some hue regions that are very small and consequently contain few if any color samples. Therefore, the optimization that we use is further constrained—we enforce that each hue region has at least five patches and that its hue angle is of at least five degrees. The optimization has been implemented using the `fmincon` function of MATLAB and used the interior point algorithm. The requirement of a minimum hue angle of each region can be expressed as a linear inequality constraint and the minimum number of patches in each region as a nonlinear inequality constraint.

Most recently, Andersen and Connah published a paper proposing another color correction method that is also hue plane preserving and based on the original HPPCC [27]. The differences between our method and their method are as

follows. In our method, a hue angle determines a single color correction matrix that has been optimized for the relevant hue region and the C0 continuity of the color correction transform is enforced using the hue region boundary constraints as discussed above. Andersen and Connah proposed to train as many color correction matrices as there are training samples and use their weighted (with respect to the hue angle of the sample to be corrected) average as the color correction matrix for that sample. Unlike our method, the resulting color correction transform is differentiable, but on the other hand it requires more calculations to determine a color correction transform for each sample.

#### 4. EXPERIMENTS AND DISCUSSION

We evaluated the performance of our new method by performing both the synthetic data simulations and a real camera experiment. The synthetic data experiments are described in the first three subsections. In Section 4.A, we describe an initial experiment with an aim to select an optimal number of hue regions providing the proposed algorithm settings for subsequent experiments. In Section 4.B, we compare the results produced by our new method to some existing state-of-the-art color correction methods. In Section 4.C, we provide additional color correction results for an extended set of camera sensors and illuminants. Finally, in Section 4.D, we perform an experiment using data acquired from two real cameras.

##### A. Selection of the Number of Hue Regions

The aim of the first experiment is to establish the influence of the number of hue regions on the performance of the proposed method.

In this experiment, we used the Sony DXC-930 [28], Foveon [29], and Nikon 5100 sensor sensitivities [7]. We employed three reflectance datasets: the X-rite SG color chart comprising 96 reflectances (shown in Fig. 1), the Macbeth DC chart comprising 180 patches, and the Simon Fraser University (SFU) reflectance dataset comprising 1995 reflectances [28].

The reflectance, illuminant D65 [6], and sensor sensitivity spectra were sampled every 10 nm from 400 to 700 nm. The above spectra were used to numerically calculate the corresponding sets of camera responses ( $RGBs$ ) and  $XYZs$ . For each of the three datasets, we learn color correction models, which are then evaluated using the leave-one-out cross validation (the smaller SG and DC reflectance datasets), i.e., for the dataset containing  $n$  samples, and we build a model from all but one sample, which is later used for testing. This is repeated  $n$  times, and the mean  $\Delta E$  in the CIELUV color space [6] is calculated. In the experiments involving the largest (SFU) dataset, instead of leave-one-out we use the 100-fold cross validation.

In the original HPPCC method, the authors used twelve hue regions. Here, we varied this number from 1 to 10. We calculated the mean, median, and 95 percentile  $\Delta E$  errors for the three datasets and the three sensor sets. The results of this experiment can be seen in Figs. 8–10. They reveal a decreasing trend of color correction errors as the number of hue regions increases. The decrease is usually clearly visible for the first few hue region partitions and is followed by a

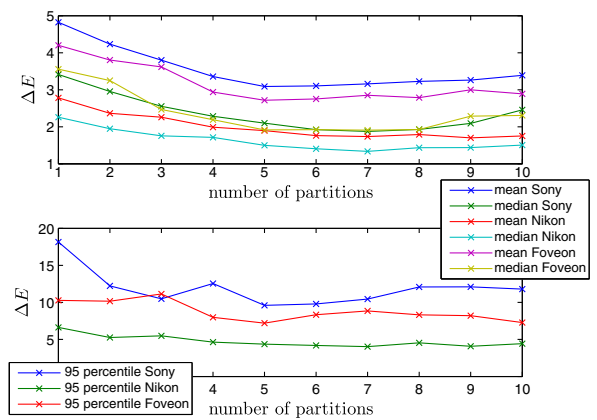


Fig. 8. Mean, median, and 95 percentile  $\Delta E$  errors for the increasing number of hue partitions for the SG chart dataset.

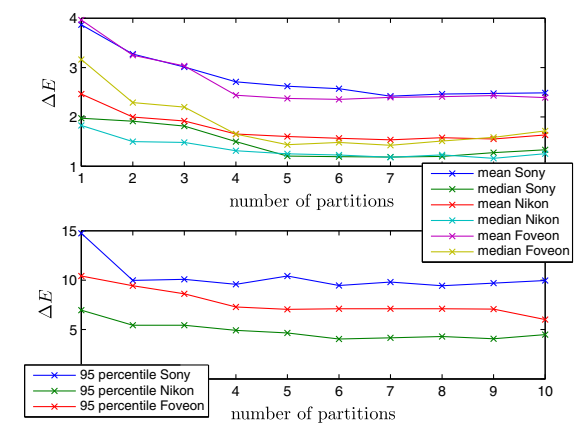


Fig. 9. As in Fig. 8 but for the DC chart.

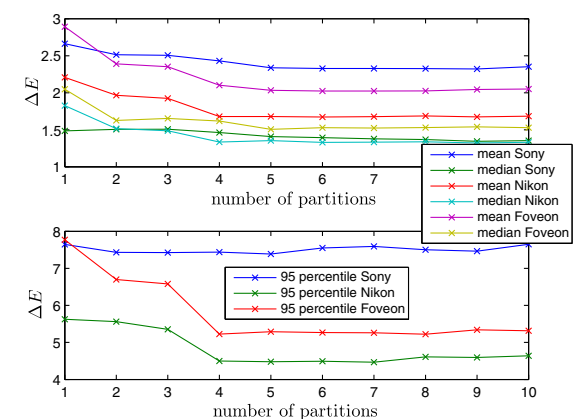


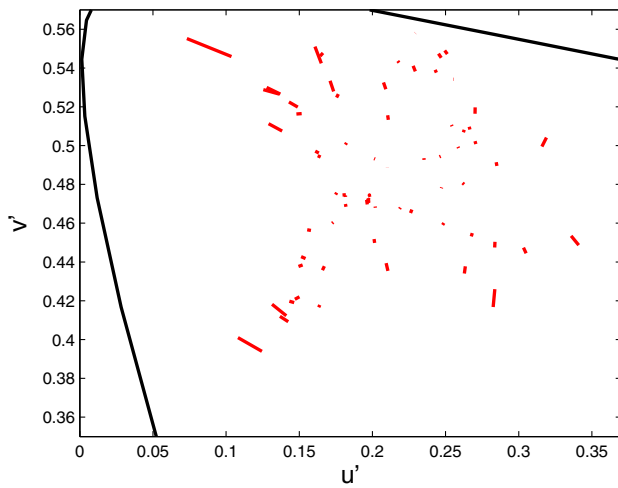
Fig. 10. As in Figs. 8 and 9 but for the SFU dataset.

plateau, with occasional small error increases. The strength of this decrease depends on the dataset and the sensor set used but is present in all of them. The errors diminish most significantly for the smaller SG and DC datasets. As for the largest SFU dataset, there is a more modest decrease in mean and

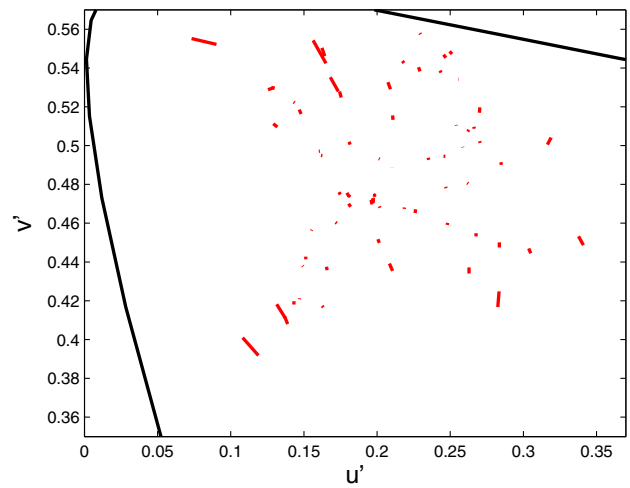
median error statistics. This difference is not, in our view, attributable to one reason alone. One factor to consider is the relative sizes of the datasets. The color chart data has fewer patches, so it is not unsurprising that we can arrive at a better fitting model. Also related to the small patch number, we carried out leave-one-out cross validation for this dataset (but not for the SFU set). Assuming a dataset has a large number of duplicates, leave-one-out cross validation tends to overfit the data, i.e., you can achieve better fitting errors but at the cost of a model that is—in effect—overfitted to the data. However, in our experiments it is important to note that we only used the distinct colors (and mitigate the possibility of overfitting). Another important factor is the distribution of the color samples. Although the SFU dataset is large, the preponderance of the data is desaturated. As such, the error (for all methods is less) and the potential for improvement is, concomitantly, less too. Having said that, for the SFU dataset, we can see a strong decrease in the 95 percentile error at least for two of the three

sensors. This figure is as or even more important than the other two error statistics as it is usually the high errors greater than  $10\Delta E$  affecting a small number of saturated samples that produce strong visible effects. Another observation that we can make is that there is little benefit in having more than six hue regions and often having as few as four provides performance that is not much below the optimum. Therefore, for further experiments in the following sections, we have decided to test our method with these two settings: four and six hue regions. We will denote these two methods as NHPPCC-4 and NHPPCC-6.

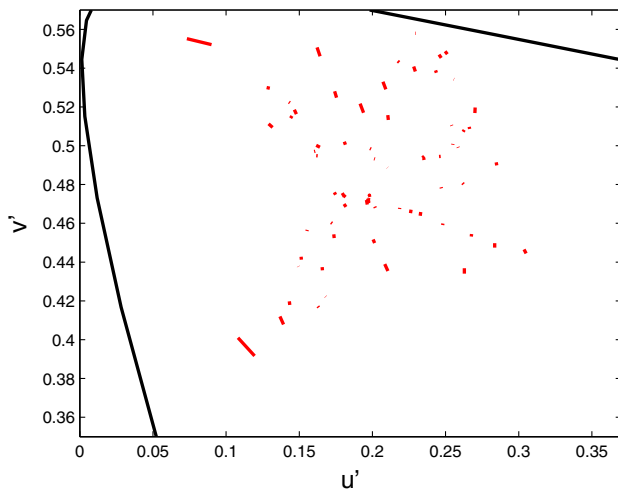
In Figures 11(a)–11(d), we can see the distribution of errors in the CIELUV chromaticity diagram for linear color correction and the proposed method with two, four and six hue regions. This result was produced for the Nikon sensor, the D65 illuminant, and the DC chart reflectance dataset. We can see that the red lines representing color correction errors generally become shorter as the number of hue regions



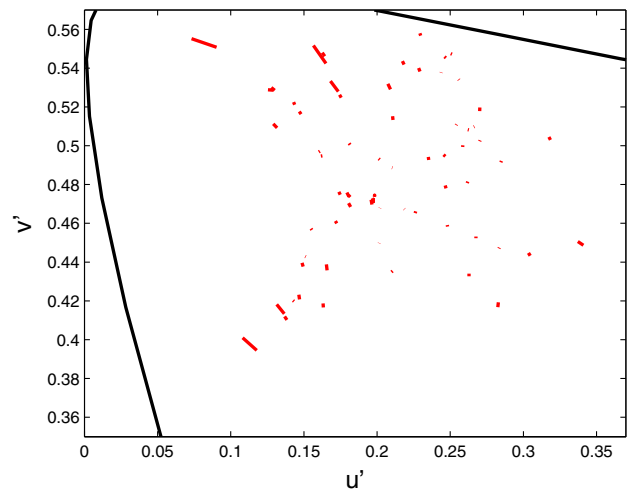
(a) LCC



(b) NHPPCC-2



(c) NHPPCC-4



(d) NHPPCC-6

**Fig. 11.** Color correction errors shown in the CIELUV chromaticity diagram. The errors plotted were calculated for the Nikon sensor, DC chart reflectance set, and four different color correction methods.



increases. This said, Figs. 11(c) and 11(d) do not suggest there is a significant performance difference between NHPPCC-4 and NHPPCC-6.

## B. Comparison with Other Color Correction Methods

In this section, we compare the NHPPCC method to the linear color correction and also to the original HPPCC. For the former, we used the hue region settings established in Section 4.A. For the latter, the hue circle was divided into twelve slices and the sample selection was performed based on a relative susceptibility to noise (for details see [22]).

Further, we compared the performance of the above to the polynomial and root-polynomial color correction methods up to degree of three [21]. An additional color correction method that was used for comparison was the tri-linear LUT interpolation with the graph Hessian regularizer [30]. The size of the LUT that we used was  $13 \times 13 \times 13$ .

We performed the above comparisons for the same three sensor sets and the three reflectance datasets that we utilized in the previous section. Analogously, we used the D65 daylight illuminant. Also, the cross validation was performed in the same way as described above. The results of these experiments can be seen in Table 1. We present the same three statistics of  $\Delta E$  errors in the CIELUV color space. The best performing algorithm in each column is shown in bold font.

We can make the following observations from these results. We can see that both NHPPCC-4 and NHPPCC-6 always perform better than the LCC and the original HPPCC. The advantage of the new method over the HPPCC is particularly visible for the largest SFU dataset as well as for all the Foveon results. This was to be expected as the HPPCC SFU dataset results expose the fact that this method was using only a small number of reflectances of this very large dataset (in fact  $12 + 1 = 13$ ) to calculate the color correction transform. Therefore, its results had to be worse than the results of all the other methods that utilized all (putting aside cross validation) dataset samples. The Nikon sensor SFU dataset results are an exception here, as the HPPCC returns a result that is comparable to the LCC. This may be due to the fact that the Nikon sensor is more colorimetric than the other two and in general all the error figures are much lower than for the other two sensors. The NHPPCC improves on HPPCC by enforcing that all color samples are used in the model fit. At the same time, it maintains the flexibility of the HPPCC in optimizing color correction in each hue region separately.

As for the second point regarding the Foveon sensor results, the clear improvement of the NHPPCC methods over HPPCC stems from the fact that the responses of this sensor set are desaturated and hence require more “aggressive” color correction. The desaturated samples that are relatively close to the

**Table 1. Synthetic Data Characterization Results<sup>a</sup>**

Dataset	SG			DC			SFU		
	Mean	Med	95 Pt.	Mean	Med	95 Pt.	Mean	Med	95 Pt.
Sony									
LCC	4.8	3.2	18	3.9	2.1	14	2.6	1.4	7.5
PCC,2	4.0	2.7	11	3.1	1.8	10	2.4	1.4	7.2
PCC,3	3.2	2.2	7.5	2.4	1.3	7.7	1.9	1.2	6.5
LUT	2.7	1.9	<b>7.3</b>	<b>1.8</b>	<b>1.0</b>	<b>6.7</b>	<b>1.5</b>	<b>1.0</b>	<b>5.0</b>
RPCC,2	2.8	1.8	8.7	2.4	1.3	8.5	2.1	1.2	7.0
RPCC,3	<b>2.6</b>	<b>1.6</b>	8.1	2.2	1.3	7.5	1.8	1.2	6.1
HPPCC	3.8	2.2	14	2.8	1.4	10	3.6	1.8	14
NHPPCC-4	3.4	2.3	13	2.7	1.5	9.6	2.4	1.5	7.4
NHPPCC-6	3.2	2.0	10	2.6	1.2	9.5	2.3	1.4	7.6
Nikon									
LCC	2.7	2.2	6.3	2.4	1.6	6.8	2.0	1.5	5.8
PCC,2	2.2	1.7	5.2	1.8	1.2	5.0	1.7	1.2	4.6
PCC,3	1.7	1.2	4.2	1.4	1.0	4.0	1.4	1.1	3.9
LUT	<b>1.5</b>	<b>1.1</b>	4.7	<b>1.1</b>	<b>0.9</b>	<b>3.1</b>	<b>1.3</b>	<b>0.9</b>	<b>3.6</b>
RPCC,2	1.7	1.4	4.3	1.5	1.1	4.2	1.5	1.1	4.2
RPCC,3	<b>1.5</b>	<b>1.1</b>	<b>3.9</b>	1.3	0.9	3.6	1.4	1.1	4.0
HPPCC	2.7	1.9	8.6	2.0	1.5	5.6	2.0	1.6	5.7
NHPPCC-4	2.0	1.7	4.6	1.7	1.3	4.9	1.7	1.3	4.5
NHPPCC-6	1.8	1.4	4.2	1.6	1.2	4.0	1.7	1.3	4.5
Foveon									
LCC	4.0	3.6	10	3.6	2.5	10	2.5	1.6	7.3
PCC,2	2.9	2.4	6.8	2.9	2.2	8.6	2.2	1.7	5.7
PCC,3	2.2	1.6	5.8	<b>2.1</b>	1.6	<b>6.3</b>	<b>1.9</b>	1.5	<b>5.0</b>
LUT	3.4	2.7	9.9	2.7	1.7	8.7	2.0	<b>1.4</b>	5.2
RPCC,2	3.0	2.3	7.1	2.9	1.8	7.0	2.0	<b>1.4</b>	5.5
RPCC,3	<b>2.0</b>	<b>1.6</b>	<b>5.1</b>	2.2	<b>1.4</b>	6.4	<b>1.9</b>	<b>1.4</b>	<b>5.0</b>
HPPCC	4.9	2.7	19	5.4	3.4	18	3.3	2.2	9.2
NHPPCC-4	2.9	2.2	8.0	2.4	1.6	7.3	2.1	1.6	5.2
NHPPCC-6	2.8	1.9	8.3	2.4	1.5	7.1	2.0	1.5	5.3

<sup>a</sup>The errors obtained are given as the mean, median, and 95 percentile error in the CIELUV color space.

achromatic center pose a problem for the HPPCC method. For that method, the hue region boundary sample selection procedure works in such a way that if the color samples are desaturated, it is difficult to choose the right color samples that would define in an optimal way the shape of the color correction transform in each region of the hue circle.

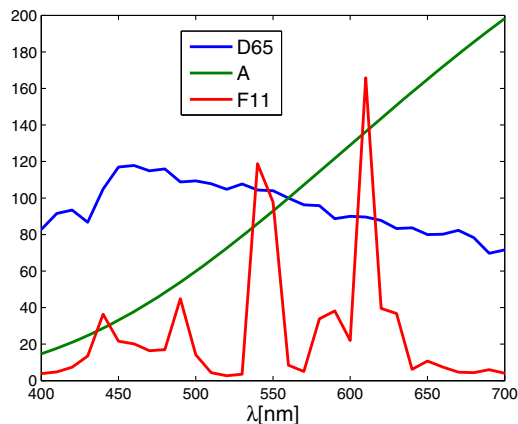
The above results also show that the results of our new method are in most cases comparable or not much worse than the state-of-art root-polynomial color correction (RPCC). The smallest errors are usually produced by the 3D LUT method or the polynomial or root-polynomial color correction of higher degrees. We must remember, however, that the LUT and polynomial methods have drawbacks. These include the lack of exposure invariance and the lack of hue plane preservation. We note that the root-polynomial color correction is invariant to exposure but does not preserve hue planes. From this point of view, it is the linear color correction, which is the benchmark of our proposed method. Our proposed method significantly outperforms this benchmark. Importantly, the NHPPCC method could also plausibly be incorporated in existing camera processing pipelines.

**C. Additional Illuminants and Sensor Sets**

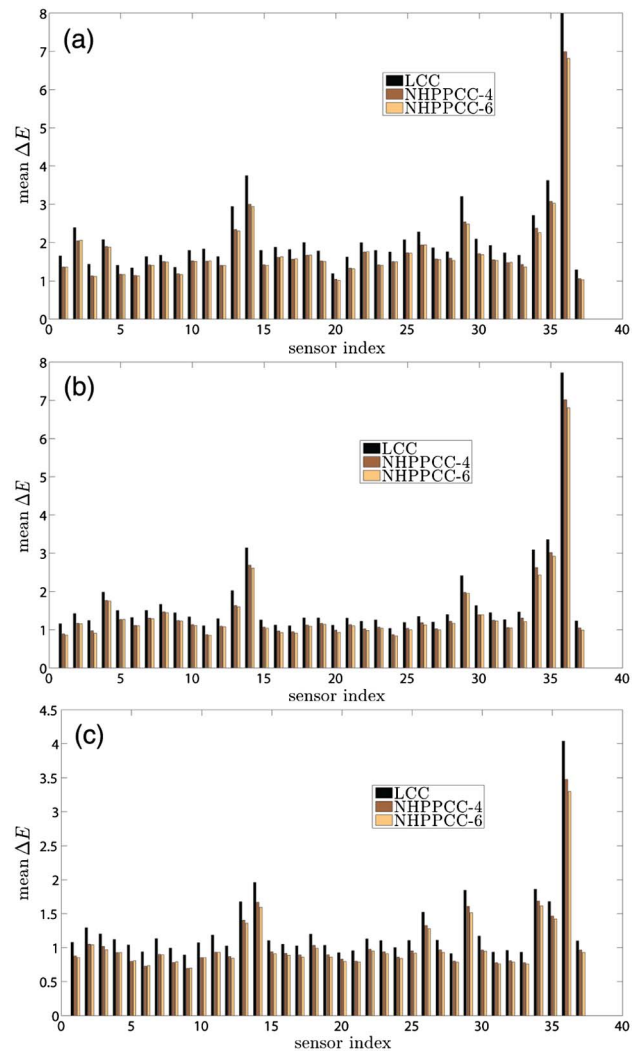
In order to confirm the observations made in the previous experiments, we carried out an additional experiment involving a significantly larger set of camera sensors. Here, we used the same dataset of 37 sensor sets that was used in [18]. The dataset includes 28 sensors sets used in Ref. [31] and nine from the Image Engineering website [32]. We also tested our method with three illuminant spectra: daylight D65, illuminant A, and fluorescent illuminant F11 (see Fig. 12). Here, we tested only the most relevant and challenging SFU reflectance dataset.

The results of this experiment are visualized in Fig. 13 and further summarized in Table 2. Each of the three figure panels contains mean  $\Delta E$  bars for all 37 sensors and three methods that we test here: linear color correction and our proposed method with the number of hue regions set to four and six. Table 2 contains the average of these mean  $\Delta E$  errors as well as average of median and 95 percentile errors.

The first observation we make is that the trends from the earlier experiments are confirmed. Our proposed method improves color correction performance comparing to the linear



**Fig. 12.** Spectra of three illuminants: D65, A, and F11.



**Fig. 13.** Color correction results for the set of 37 sensors. SFU reflectance dataset. (a) Illuminant D65, (b) A, and (c) F11.

color correction for all combinations of 37 sensor sets and three illuminants. Next, we can see that the improvement is most significant for the least colorimetric sensors, where the scope for improvement is the widest. We can also see only a very small advantage of NHPPCC-6 over NHPPCC-4. Again, this advantage is the most visible for the least colorimetric sensors. Importantly, we observe significant improvements of average 95 percentile errors for all three illuminants.

**D. Real Data Experiments**

In this section, we describe an experiment, which unlike all previous experiments, was performed on a dataset captured with cameras. We used the two real camera datasets (Nikon D70 and Sigma SD15) that we described earlier in Ref. [21]. Both datasets have been captured as follows. We placed the X-rite SG color checker in a viewing box, which was illuminated by a D65 simulator [6]. The illumination was provided by a Gamma Scientific RS-5B LED illuminator [33]. The scene containing the color checker was imaged with both cameras set to RAW image capture mode. A Photo Research PR-650

**Table 2. Average Results for 37 Sensors and Three Illuminants**

Model Type	D65			A			F11		
	Mean	Med	95 Pt.	Mean	Med	95 Pt.	Mean	Med	95 Pt.
LCC	2.3	1.7	6.2	1.7	1.2	5.0	1.3	0.8	3.7
NHPPCC-4	<b>1.8</b>	<b>1.3</b>	5.0	1.5	<b>1.0</b>	4.3	1.1	<b>0.7</b>	3.2
NHPPCC-6	<b>1.8</b>	<b>1.3</b>	<b>4.9</b>	<b>1.4</b>	<b>1.0</b>	<b>4.2</b>	<b>1.0</b>	<b>0.7</b>	<b>3.1</b>

**Table 3. Nikon D70 and Sigma SD15 Characterization Results<sup>a</sup>**

Model Type	Nikon D70			Sigma SD15		
	Mean	Med	95 Pt.	Mean	Med	95 Pt.
LCC	2.5	2.3	5.1	5.2	4.0	16
PCC,2	2.1	1.8	4.8	3.9	3.2	9.4
PCC,3	1.7	1.6	<b>3.1</b>	<b>3.1</b>	<b>2.3</b>	<b>7.0</b>
LUT	1.7	1.4	4.4	4.3	3.3	12
RPCC,2	1.9	1.7	4.0	3.8	2.8	8.7
RPCC,3	<b>1.6</b>	<b>1.3</b>	3.4	3.2	2.4	9.1
HPPCC	2.5	2.1	6.0	5.0	3.2	17
NHPPCC-4	2.0	1.8	4.3	3.9	3.0	10
NHPPCC-6	1.9	1.7	3.9	3.5	2.6	9.1

<sup>a</sup>The error statistics are given as in Table 1.

spectrophotometer was used to measure the  $XYZs$  of the 96 patches of the color checker. We used DCRAW [34] (Nikon) and PROXEL X3F [35] (Sigma) to extract the 16 bit linear images. We calculated the average RGB of each manually segmented color checker patch. The dark frames were captured with the lens cap on and then subtracted from the average camera responses. We did not attempt to correct for the camera optics (which might also affect intensity at the pixel level). We used the average  $RGB$  responses and the measured  $XYZs$  to color characterize the two cameras according to all the models described in Section 4.B. The models were also validated and evaluated in the same way as those derived from the synthetic data. The results of these experiments are presented in Table 3.

We can see that the results very closely follow the trends that we could see in the synthetic data experiments. The NHPPCC methods are always better than the linear color correction as well as the original HPPCC. They also provide a comparable performance to the remaining methods, including root-polynomial color correction. As for the synthetic data, the improvement in performance is the most significant for the camera that initially produced the worst results.

## 5. CONCLUSIONS

Hue plane preserving color correction using constrained least squares is based on the earlier algorithm that utilized a small set of linear corrections where the correction chosen depended on hue. Our extensive experiments prove that unlike its predecessor, our new method is robust. It provides a step change in performance over linear color correction for all tested cameras and datasets. The method benefits from being exposure invariant as well as preserving hue planes. It provides results that are only marginally worse than those provided

by state-of-art root-polynomial color correction and LUT-based color correction. While these nonlinear methods often offer a small improvement over our method, they do exhibit drawbacks: a lack of exposure invariance (LUT, PCC) and a lack of hue preserving (LUT, PCC, RPCC). Our method retains advantages of both nonlinear and linear color correction methods. The former arise from that our method is globally nonlinear and hence it offers a significant correction improvement over LCC, the latter in essence stem from the fact that our nonlinear color correction method is piecewise linear, i.e., it consists of a set of linear transforms each operating in its own convex cone in the color space. Therefore, in addition to being exposure invariant and hue preserving, our method will have the same good noise characteristics as the standard linear color correction.

Supplementary material is available at [36].

**Funding.** Engineering and Physical Sciences Research Council (EPSRC) (EP/J005223/1).

**Acknowledgment.** This work was supported by the UK Engineering and Physical Sciences Research Council (grant number EP/J005223/1). The authors would like to thank Javier Vazquez-Corral for his help in collating the sensor spectra database used in the experiments in Section 4.C.

## REFERENCES

1. G. Wyszecki and W. Styles, *Color Science: Concepts and Methods, Quantitative Data and Formulae* (Wiley, 1982).
2. M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, "A standard default color space for the internet—sRGB," Technical Report (Hewlett-Packard, Microsoft, 1996).
3. R. T. D. Luther, "Aus dem Gebiet der Farbreizmetrie," *Z. Tech. Phys.* **8**, 540–555 (1927).
4. H. E. Ives, "The transformation of color-mixture equations from one system to another," *J. Franklin Inst.* **180**, 673–701 (1915).
5. P. Hubel, J. Holm, G. Finlayson, and M. Drew, "Matrix calculations for digital photography," in *Proceedings of the 5th Color and Imaging Conference (CIC)* (1997).
6. R. Hunt and M. R. Pointer, *Measuring Colour*, 4th ed. (Wiley, 2011).
7. M. M. Darrodi, G. Finlayson, T. Goodman, and M. Mackiewicz, "Reference data set for camera spectral sensitivity estimation," *J. Opt. Soc. Am. A* **32**, 381–391 (2015).
8. H. Kang, "Colour scanner calibration," *J. Imaging Sci. Technol.* **36**, 162–170 (1992).
9. R. S. Berns and M. J. Shyu, "Colorimetric characterization of a desktop drum scanner using a spectral model," *J. Electron. Imaging* **4**, 360–372 (1995).
10. G. Hong, M. R. Luo, and P. A. Rhodes, "A study of digital camera characterisation based on polynomial modelling," *Color Res. Appl.* **26**, 76–84 (2001).
11. T. Cheung and S. Westland, "Colour camera characterisation using artificial neural networks," in *10th Color Imaging Conference* (The Society for Information Display, 2002), vol. **4**, pp. 117–120.

12. H. Kang and P. Anderson, "Neural network application to the color scanner and printer calibration," *J. Electron. Imaging* **1**, 125–134 (1992).
13. L. Xinwu, "A new color correction model based on bp neural network," *Adv. Inf. Sci. Serv. Sci.* **3**, 72–78 (2011).
14. T. Cheung, S. Westland, D. Connah, and C. A. Ripamonti, "Comparative study of the characterization of colour cameras by means of neural networks and polynomial transforms," *J. Color. Technol.* **120**, 19–25 (2004).
15. P. Hung, "Colorimetric calibration in electronic imaging devices using a look-up tables model and interpolations," *J. Electron. Imaging* **2**, 53–61 (1993).
16. J. McElvain and W. Gish, "Camera color correction using two-dimensional transforms," in *Proceedings of the 21st Color and Imaging Conference (CIC)* (2013).
17. S. H. Lim and A. Silverstein, "Spatially varying colour correction matrices for reduced noise," Technical Report HPL-2004-99 (Imaging Systems Laboratory, HP Laboratories, 2004).
18. J. Vazquez-Corral, D. Connah, and M. Bertalmio, "Perceptual color characterization of cameras," *Sensors* **14**, 23205–23229 (2014).
19. G. D. Finlayson and M. Drew, "Constrained least-squares regression in color spaces," *J. Electron. Imaging* **6**, 484–493 (1997).
20. P. Green and L. W. MacDonald, eds., *Colour Engineering: Achieving Device Independent Colour* (Wiley, 2002).
21. G. Finlayson, M. Mackiewicz, and A. Hurlbert, "Colour correction using root-polynomial regression," *IEEE Trans. Image Process.* **24**, 1460–1470 (2015).
22. C. F. Andersen and J. Y. Hardeberg, "Colorimetric characterization of digital cameras preserving hue planes," in *Proceedings of the 13th Color and Imaging Conference (CIC)* (2005), pp. 141–146.
23. B. T. Phong, "Illumination for computer generated pictures," *Commun. ACM* **18**, 311–317 (1976).
24. M. Mackiewicz, C. Andersen, and G. Finlayson, "Hue plane preserving colour correction using constrained least squares regression," in *Proceedings of the 22nd Color and Imaging Conference (CIC)* (2015).
25. A. Laratta and F. Zironi, "Computation of lagrange multipliers for linear least squares problems with equality constraints," *Computing* **67**, 335–350 (2001).
26. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems* (SIAM, 1995).
27. C. F. Andersen and D. Connah, "Weighted constrained hue plane preserving camera characterization," *IEEE Trans. Image Process.* **25**, 4329–4339 (2016).
28. K. Barnard, L. Martin, B. Funt, and A. Coath, "A dataset for color research," *Color Res. Appl.* **27**, 147–151 (2002).
29. P. Hubel, "Foveon technology and the changing landscape of digital cameras," in *Proceedings of the 13th Color and Imaging Conference (CIC)* (2005), pp. 314–317.
30. E. Garcia, R. Arora, and M. R. Gupta, "Optimized regression for efficient function evaluation," *IEEE Trans. Image Process.* **21**, 4128–4140 (2012).
31. J. Jiang, D. Liu, J. Gu, and S. Susstrunk, "What is the space of spectral sensitivity functions for digital color cameras?" in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)* (IEEE Computer Society, 2013), pp. 168–179.
32. "Spectral sensitivities of several cameras," <http://www.image-engineering.de/>.
33. M. Mackiewicz, S. Crichton, S. Newsome, R. Gazerro, G. Finlayson, and A. Hurlbert, "Spectrally tunable led illuminator for vision research," in *Proceedings of the 6th Colour in Graphics, Imaging and Vision (CGIV)* (2012), Vol. **6**, pp. 372–377.
34. "DCRAW: Decoding raw digital photos in Linux," <https://www.cybercom.net/~dcoffin/dcraw/>.
35. "PROXEL X3F Tools," <http://www.proxel.se/x3f.html>.
36. [http://research.cmp.uea.ac.uk/hppcc\\_sup\\_material/](http://research.cmp.uea.ac.uk/hppcc_sup_material/).