

Designing a Color Filter via Optimization of Vora-Value for Making a Camera more Colorimetric

Yuteng Zhu*, Graham D. Finlayson
School of Computing Sciences, University of East Anglia, Norwich, UK
* yuteng.zhu@uea.ac.uk

Abstract

Previous work has proposed to solve for a filter which, when placed in front of a camera, improves the colorimetric property by best satisfying the Luther condition. That is, the filtered spectral sensitivities of a camera - after a linear transform - are as close to the color matching functions of the human visual system as possible. By construction, the prior art solves for a filter for a given set of human visual sensitivities, e.g. the XYZ color matching functions or the cone response functions. However, depending on the target spectral sensitivity set, a different optimal filter is found.

In this paper, we set out a method to solve for a filter that works equally well for all possible target sensitivity sets of the human visual system. We observe that the cone fundamentals, the CIE XYZ color matching functions or any linear combination thereof, span the same vector space. Thus, we solve for a filter that makes the vector space spanned by the filtered camera sensitivities as similar as possible to the space spanned by human vision sensors. We argue that the Vora-Value is a suitable way to measure subspace similarity and we develop an optimization method for finding a filter that maximizes the Vora-Value measure.

Experiments demonstrate that our new optimization leads to the filtered camera sensitivities which have a significantly higher Vora-Value and improved colorimetric performance compared with antecedent methods.

1. Introduction

Color cameras are not colorimetric. That is, they do not satisfy the Luther condition [1]: their spectral sensitivities are not a linear transform from the CIE XYZ color matching functions (or equivalently any linear combination of the CMFs) [2]. Only cameras that satisfy the Luther condition can measure colors in all conditions which can then be corrected, without error, to colorimetric counterparts [3].

There are many papers in the literature address the colorimetric problem by proposing methods that design optimal filters to accurately recover tristimulus color responses [4–11]. Recently, Finlayson *et al.* [12] developed a method for finding the best transmissive filter that, when placed in front of a camera (visualised in Figure 1), results in the new effective sensitivities that best meet the Luther condition. Curiously, however, if we change the target sensitivity functions - e.g. from XYZ CMFs to the cone fundamentals, themselves a linear transform apart [13], the solved-for best filter changes.

In this paper, we develop a method for finding a filter which makes the vector space spanned by the effective camera spectral

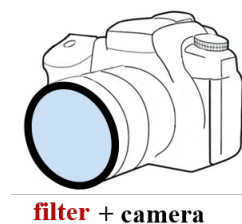


Figure 1: Illustration of the idea of placing a color filter in front of a camera for measurement.

sensitivities closest to all possible target sensitivity function sets for the Human Visual Space (HVS). We illustrate our approach in Figure 2. In blue, we show the vector space spanned by the sensitivity functions of the HVS. These sensitivities might be the cone fundamentals or equivalently the XYZ CMFs (we say equivalently because these sensitivity sets are linearly related and both sets span the same subspace). Similarly, the camera spectral sensitivities - shown in red - span a different vector space.

When we place a filter in front of a camera, the new effective spectral sensitivities span a new subspace, shown in green in Figure 2. In this paper, we set forth a method for finding a filter which makes the filtered sensitivities spanning a subspace (the green vector space) as close as possible to the human visual space (the blue vector space).

Let's start to make some of this intuition more concrete. First, let us introduce a little notation. We denote respectively the spectral sensitivities of the camera and the human visual sensors as \mathbf{Q} and \mathbf{X} . For the purposes of this paper, we can think of these sensor sets as tables of discrete measurements. The columns of \mathbf{Q} are respectively the spectral sensitivities of the Red, Green and Blue sensors. Each row of \mathbf{Q} is the three camera responses at discrete wavelengths. Usually, we measure sensitivities from 400 nm to 700 nm (the typical wavelength range of visible spectrum) with a 10 nm sampling distance, so the spectral sensitivity tables (or matrices) have 31 rows. \mathbf{X} is similarly defined where columns could be the XYZ color matching functions or any function set thereof.

The blue and red planes in Figure 2 - for the human visual system and camera sensitivities, respectively - visualise the subspaces spanned by \mathbf{Q} and \mathbf{X} . By span we mean the set of all spectra we can generate by taking linear combinations of the columns of \mathbf{Q} (or \mathbf{X}). We call a subspace because the columns of \mathbf{Q} (or \mathbf{X}) span a 3-dimensional vector space contained in the 31-dimensional

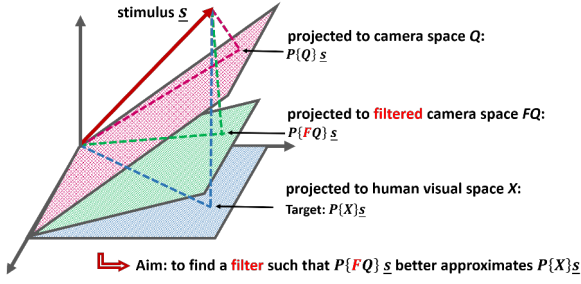


Figure 2: A visualisation of the subspaces of camera space (in red), filtered camera space (in green), and the human visual space (in blue). We aim for the camera space with a color filter to be as close as possible to the human visual space.

space. So, how do we evaluate the closeness (or otherwise) of two subspaces spanned by the column vectors of \mathbf{Q} and \mathbf{X} ?

Usually, when we compare quantities, they are in *correspondence*. So far the ideas of vectors spanning a subspace has been quite abstract. The key to comparing subspace is defining a method of placing spectra generated in one subspace in correspondence to matching spectra in another. Let us generate a 31-dimensional spectrum \underline{s} . Returning to Figure 2, we denote the projection of \underline{s} onto \mathbf{Q} as $P\{\mathbf{Q}\}\underline{s}$. The corresponding projection onto \mathbf{X} is denoted $P\{\mathbf{X}\}\underline{s}$. Finally, we denote the spectral sensitivities of a camera with a filter, \mathbf{F} , placed in front as \mathbf{FQ} and the corresponding projected spectrum as $P\{\mathbf{FQ}\}\underline{s}$. For the stimulus \underline{s} , now we find the corresponding (projected) spectra, $\underline{s}_{\mathbf{Q}}$ and $\underline{s}_{\mathbf{X}}$ in the vectors spaces spanned by \mathbf{Q} and \mathbf{X} that are closest to \underline{s} . And, given these corresponding spectra, we can calculate their distance: $\|\underline{s}_{\mathbf{Q}} - \underline{s}_{\mathbf{X}}\|_2$. By generating lots of random spectra - the values of which are independent and identically distributed - drawn from 31-dimensional space, we can calculate the mean difference between their *projections* to the spaces spanned by \mathbf{Q} and \mathbf{X} . This projected spectral error (after normalization) - calculated over all possible spectra - is our definition of subspace similarity.

This reasoning leads to a simple, closed-form, subspace similarity formula that calculates a number in $[0,1]$. Intuitively, it is kind of a percentage measure of how similar two subspaces are. Actually, to this point our reason is (perhaps with a slightly different emphasis) a retelling of the derivation of the Vora-Value [14]. The Vora-Value is a measure of how well a camera's RGBs can be (linearly) corrected to colorimetric counterparts (under the so-called Maximum Ignorance Assumption [15] which posits that all spectra are equally likely). Equivalently, it can be thought of as a percentage type measure of the similarity of two vector spaces, here the space spanned by two sensor sets (a camera and the HVS).

In this paper, we formulate an optimization expression using the Vora-Value formula which we can maximize to find the *best* color filter to place in front of a camera (i.e. the one that returns the best subspace similarity). The optimization method itself is not closed-form but we show we can differentiate the objective function (with respect to the filter we seek) and then find the filter by a gradient ascent method [16].

Experiments demonstrate that the Vora-Value score for many cameras is increased significantly by the optimized filters returned by our optimization. Further, the current method gives higher Vora-Values and consequently lower color measurement errors than when the filters are solved by the prior Luther-condition

method [12].

In this paper, Section II reviews the definition of Vora-Value in detail. We maximize the Vora-Value by a gradient ascent approach in Section III. The experimental results and discussions are given in Section IV. The paper concludes in Section V.

2. Background

Vora-Value

The Vora-Value of a camera sensor set \mathbf{Q} and the XYZ color matching functions \mathbf{X} is defined [17] as

$$v(\mathbf{Q}, \mathbf{X}) = \frac{\text{trace}(P\{\mathbf{Q}\}P\{\mathbf{X}\})}{\text{trace}(P\{\mathbf{X}\})} \quad (1)$$

where $P\{\mathbf{Q}\}$ and $P\{\mathbf{X}\}$ denote the projection matrices of the camera sensitivity functions and the color matching functions, respectively and $\text{trace}()$ is the trace of a square matrix that sums up the elements along the diagonal of a matrix. In the equation, $\text{trace}(P\{\mathbf{X}\})$ is a normalization term and so the Vora-Value is in the range of $[0, 1]$. For a trichromatic vision system where \mathbf{X} has three independent sensitivity functions, then $\text{trace}(P\{\mathbf{X}\}) = 3$ (the trace of a projector is equal to the rank of the matrix).

The best case is when the subspace spanned by the camera sensitivities is the same as the subspace spanned by the color matching functions then $P\{\mathbf{X}\} = P\{\mathbf{Q}\}$. Noting idempotency property of projectors $P\{\mathbf{X}\}P\{\mathbf{X}\} = P\{\mathbf{X}\}$, the Vora-Value equals 1. When the camera sensitivity curves are perpendicular to - in the null space of - the color matching functions, i.e. $P\{\mathbf{Q}\} = I - P\{\mathbf{X}\}$, the Vora-Value is 0. The Vora-Value can be thought of as a percentage measure of the similarity between two subspaces (see discussion in the Introduction) in general and of sets of spectral sensitivities in particular.

The projection of an $n \times m$ matrix A with rank m ($n > m$) is defined as:

$$P\{A\} = A(A^T A)^{-1} A^T \quad (2)$$

where the superscripts T and -1 denote respectively the matrix transpose and inverse. Substituting the explicit definition of a projector into Eq. 1 gives

$$v(\mathbf{Q}, \mathbf{X}) = \frac{1}{3} \text{trace}(\mathbf{Q}(\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{Q}^T \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T). \quad (3)$$

Later we will show how we can find a filter that when placed in front of a camera results in effective camera sensitivities that yield a much higher Vora-Value (compared to the unfiltered camera sensitivities). Our optimization is solved iteratively: we follow the gradient calculated from an optimization statement.

3. Problem Statement and Algorithm

Filter-modified Vora-Value

When a filter is placed in front of a camera, it can be modeled as the multiplication of the filter spectral transmittance and the camera spectral sensitivities. On a per wavelength basis, let $[r(\lambda), g(\lambda), b(\lambda)]$ and $f(\lambda)$ denote respectively the camera spectral sensitivities and the filter transmittance at a sampled wavelength. Then the new effective sensitivities of the filtered camera are equal to $[f(\lambda)r(\lambda), f(\lambda)g(\lambda), f(\lambda)b(\lambda)]$.

For our optimization it is useful to represent spectral quantities as vectors. The filter vector \underline{f} denotes an n -dimensional vector of transmittances across the visible spectrum. Since we typically sample the visible spectrum from 400nm to 700nm at every 10nm interval then n is 31. Together with camera spectral sensitivity matrix Q , the new effective sensitivity responses after filtering can be represented as FQ where the filter vector is converted into a diagonal matrix $F = \text{diag}(\underline{f})$ for mathematical convenience, with its i th diagonal equals to the i th element of \underline{f} , i.e. $F_{ii} = f_i$.

Thus the filter-modified Vora-Value for the effective ‘filter+camera’ system can be written as

$$v(FQ, X) = \frac{\text{trace}(P\{FQ\}P\{X\})}{\text{trace}(P\{X\})} \quad (4)$$

Remind that $\text{trace}(P\{X\}) = 3$ for a trichromatic vision system. Substituting the projector matrices as Eq. 2, we have

$$v(FQ, X) = \frac{1}{3} \text{trace}(FQ(Q^T F^2 Q)^{-1} Q^T F X (X^T X)^{-1} X^T). \quad (5)$$

Note in writing this equation we made us of the fact that $F^T = F$ (diagonal matrices are symmetric).

Constraining the Shape of the Filter

We would like to control the shape of the transmittance filter. Let us assume that a filter is a liner combination of a set of k basis filters:

$$\underline{f} = B\underline{c} \quad (6)$$

where the columns of matrix B denote the basis and \underline{c} is a $k \times 1$ vector denoting the coefficients to each associated filter basis.

By judicious choice of basis, we can effectively bound the smoothness of the filter solution. Here we use the first k terms in a Cosine basis expansion [19] – which are smooth functions.

Bounding Filter Transmittance

Physically, the transmittance of a filter should be within the range of 0 and 100% meaning fully absorptive and transmissive, respectively. However, if our algorithm finds a filter that transmits little light, it will have limited practical utility for our proposal of using the ‘filter+camera’ system in accurate color measurement.

In practice, we would like the filter transmittance in the desired bounds: $f_{\min} \preceq \underline{f} \preceq f_{\max}$. Here we use the symbol \preceq to represent the vector inequality where every element in the vector satisfies the inequality. The scalars f_{\min} and f_{\max} denote the lower and upper thresholds on the desired transmittance of the filter.

Optimization Formulation and its Derivative

The objective of our optimization is simply to find a filter \underline{f} that maximizes the Vora-Value

$$\arg \max_F v(FQ, X) \quad (7)$$

where Q and X are 31×3 matrices and F is a 31×31 diagonal matrix where the filter elements are the diagonal components as $F = \text{diag}(\underline{f})$. In the projector expanded form, the optimization is written as:

$$\arg \max_F \text{trace}(FQ(Q^T F^2 Q)^{-1} Q^T F X (X^T X)^{-1} X^T). \quad (8)$$

Strictly speaking, we should also divide by 3 (so the Vora-Value is between 0 and 1 but we ignore this step as a scalar does not affect the optimization).

The derivative of the objective function $v(FQ, X)$ with respect to the filter is captured by:

$$\frac{\partial v(FQ, X)}{\partial F_{ii}} = [X(X^T X)^{-1} X^T F Q (Q^T F^2 Q)^{-1} Q^T - F Q (Q^T F^2 Q)^{-1} Q^T F X (X^T X)^{-1} X^T F Q (Q^T F^2 Q)^{-1} Q^T]_{ii} \quad (9)$$

Note because our filter is represented by a diagonal matrix: $F = \text{diag}(\underline{f})$, we are only interested in the partial derivatives along the diagonal F . Space does not permit a full recapitulation of the steps involved in differentiating v . Mainly we simply employ well known rules of matrix derivatives. We direct the reader to [18] for a review.

Rewriting Eq. 9 using our projector matrix notation:

$$\frac{\partial v(FQ, X)}{\partial F_{ii}} = [(I - P\{FQ\})P\{X\}P\{FQ\}F^{-1}]_{ii} \quad (10)$$

Now, let us rewrite the derivative in terms of the underlying filter vector \underline{f} . First, remember that $F = \text{diag}(\underline{f})$. Here the $\text{diag}()$ function takes an n -component vector and places it along the diagonal of an $n \times n$ diagonal matrix. Let us now denote the inverse operator - the one that extracts the diagonal from a square matrix and places the result in a vector - as $\text{vecd}()$. Clearly, $\text{vecd}(\text{diag}(\underline{f})) = \underline{f}$. Please also bear in mind that $\text{vecd}() \neq \text{vec}()$. The latter is often used in the Kronecker Product extension to matrix algebra and maps an $n \times m$ matrix into an $nm \times 1$ vector [18]. Here, $\text{diag}()$ is a forward operation turning a vector into a diagonal matrix and $\text{vecd}()$ is the companion reverse operator extracting the diagonal.

We can write:

$$\frac{\partial v(\text{diag}(\underline{f})Q, X)}{\partial \underline{f}} = \text{vecd}((I - P\{FQ\})P\{X\}P\{FQ\}F^{-1}) \quad (11)$$

where the diagonal values of $(I - P\{FQ\})P\{X\}P\{FQ\}F^{-1}$ are taken to form the gradient function with respect to the filter vector \underline{f} .

When $\underline{f} = B\underline{c}$, using the chain rule, the gradient function with respect to the coefficient vector \underline{c} is calculated as:

$$\frac{\partial v(\text{diag}(B\underline{c})Q, X)}{\partial \underline{c}} = B^T \frac{\partial v(\text{diag}(\underline{f})Q, X)}{\partial \underline{f}} \quad (12)$$

or, equivalently in its explicit form as

$$\frac{\partial v(\text{diag}(B\underline{c})Q, X)}{\partial \underline{c}} = B^T \text{vecd}((I - P\{\text{diag}(B\underline{c})Q\})P\{X\}P\{\text{diag}(B\underline{c})Q\} \text{diag}(B\underline{c})^{-1}) \quad (13)$$

We make two final remarks. First, clearly the gradient has a very interesting structure: it is the diagonal of the product of 3 projection matrices multiplied by the inverse of the filter. Second, since we have an explicit gradient representation we are in control of our own maximization (see details in the next section). This is important as in future work we aim to investigate the optimization in more detail including the likelihood that there are potentially many local maxima and that these maxima could be closely located.

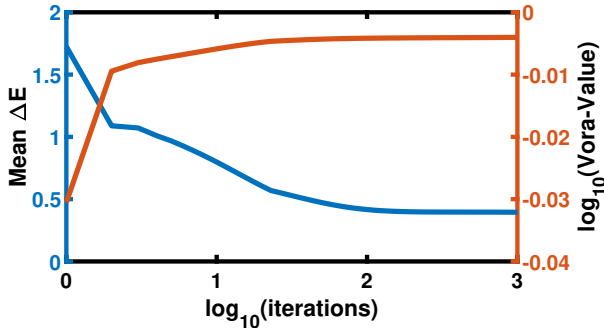


Figure 3: Algorithm convergence in terms of Vora-Value (to the right y-axis) and mean ΔE_{ab}^* with respect to iterations (to the left y-axis). Note that we plot the x-axis under \log_{10} .

Gradient Ascent

The objective function we proposed in Eq. (7) does not have a simple closed-form solution. This is where we use the gradient-based algorithm [16]. Starting with an initial guess, it iteratively steps towards a direction – the gradient of the objective function – for a solution with a smaller objective error. By repeatedly following the gradient downwards, we eventually find ourselves in a (local) minimum.

In this paper, given the gradient function (calculated in the last section), we find the best filter by gradient ascent as we will be maximizing the Vora-Value function. Details are given in Algorithm 1. An important parameter of the algorithm is the step size. The gradient vector of the function being maximized encodes the direction we should move to make the objective function larger. However, it does not teach directly how far we should move. If we add too much of the gradient we can step over the maximum and never reach the maximum point. Here the step size changes per iteration and is chosen by the backtracking line search algorithm [16].

Algorithm 1 Gradient Ascent Algorithm

- 1: $k = 0$, $\underline{f}^0 = \underline{f}^{initial}$
 - 2: **repeat**
 - 3: $k \leftarrow k + 1$
 - 4: Compute the Gradient: $\nabla v(\underline{f}^k)$
 - 5: Choose the stepsize: $t^k > 0$
 - 6: Update $\underline{f}^{k+1} = \underline{f}^k + t^k \nabla v(\underline{f}^k)$
 - 7: **until** $\max(|\nabla v(\underline{f}^k)|) \leq \eta$
 - 8: **return** \underline{f}^{k+1}
-

In Figure 3, we show how the gradient ascent algorithm converges for one camera (Canon 40D DSLR). On the x-axis we show the number of iterations (under \log_{10}) of our algorithm. For each iteration we record the Vora-Value and the mean color error. We see here that our optimization converges quickly after about 100 iterations (for both indicators).

Gradient method can also be carried out with inequality constraints. For example, in this paper we would like to bound the range of filter transmittance: $f_{min} \preceq \underline{f} \preceq f_{max}$, with respect to which we wish to find our filter. In the results shown in the next section we use the Projected Gradient Ascent Algorithm [20] for

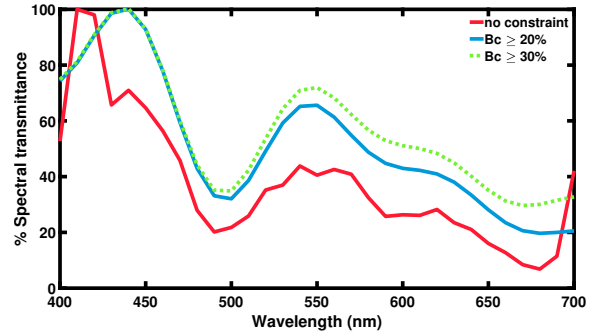


Figure 4: Spectral transmittance of Vora-Value optimized filters for Canon 40D camera. The filter in solid red line is solved with no constraint applied. Filters in solid blue and dotted green are modeled under 8-Cosine basis and constrained by minimum transmittance of 20% and 30%, respectively.

constrained filters, a simple variant of the gradient ascent with inequality constraints.

4. Experiments and Results

Vora-Value optimized filters for a Canon 40D digital camera [21] solved using Algorithm 1 are shown in Figure 4. In red is the filter found when no constraint (smoothness nor min max transmittances) is applied. In Blue, filters are described by the first 8 terms in a Cosine basis expansion and constrained to be between 20% and 100% transmissive. Finally in dotted green we increase the minimum transmission to 30%.

In Table 1, we evaluate the optimized filters using our current method in terms of Vora-Value and compare the results with the prior art of Luther condition optimization [22] under the same constraint conditions. We also include the results of the unfiltered native camera sensor as baseline results. Using the current **Vora-filter**, Vora-Value of the effective camera sensitivities (to the reference CIE1931 XYZ color matching functions) increases to 0.991 from 0.932 for its unfiltered **native** sensor sensitivities and higher than 0.986 by the **Luther-filter**, see first three rows under the column of ‘**Vora-Value**’ in Table 1. Remember that a higher Vora-Value indicates greater similarity of the subspaces spanned by the sensitivities of a camera and human visual system, therefore generally relates to more accurate color measurement [14].

We also evaluate the derived filters with respect to a color measurement experiment. For a collection of 102 illuminants and 1995 reflectance spectra [23], we calculate the RGBs (for the native camera and the camera sensitivities after filtering) and ground-truth XYZs numerically. The corresponding CIELAB color difference

Table 1: Error statistics for Canon 40D camera

Method	Vora-Value	color errors ΔE_{ab}^*				
		mean	median	95%	99%	max
Baseline	0.932	1.72	1.03	5.12	12.94	28.39
Luther-filter	0.986	0.44	0.22	1.48	3.19	8.77
Vora-filter	0.991	0.38	0.20	1.23	2.97	9.89

rection,” *IEEE Transactions on Image Processing*, vol. 3, no. 2, pp. 147–161, 1994.

- [5] M. J. Vrhel, H. J. Trussell, and J. Bosch, “Design and realization of optimal color filters for multi-illuminant color correction,” *Journal of Electronic Imaging*, vol. 4, pp. 6–14, 1995.
- [6] M. Wolski, C. Bouman, J. P. Allebach, and E. Walowit, “Optimization of sensor response functions for colorimetry of reflective and emissive objects,” *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 507–517, 1996.
- [7] P. L. Vora and H. J. Trussell, “Mathematical methods for the analysis of color scanning filters,” *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 321–327, 1997.
- [8] G. Sharma and H. J. Trussell, “Digital color imaging,” *IEEE transactions on Image Processing*, vol. 6, no. 7, pp. 901–932, 1997.
- [9] G. Sharma and H. J. Trussell, “Figures of merit for color scanners,” *IEEE transactions on Image Processing*, vol. 6, no. 7, pp. 990–1001, 1997.
- [10] W. Wu, J. P. Allebach, and M. Analoui, “Imaging colorimetry using a digital camera,” *Journal of Imaging Science and Technology*, vol. 44, no. 4, pp. 267–279, 2000.
- [11] P. L. Vora, “Inner products and orthogonality in color recording filter design,” *IEEE transactions on Image Processing*, vol. 10, no. 4, pp. 632–642, 2001.
- [12] G. D. Finlayson, Y. Zhu, and H. Gong, “Using a Simple Colour Pre-filter to Make Cameras More Colorimetric,” in *Color Imaging Conference*, no. 1, pp. 182–186, 2018.
- [13] G. Wyszecki and W. S. Stiles, *Color science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed. Wiley New York, 1982.
- [14] P. L. Vora, and H. J. Trussell, “Measure of goodness of a set of color-scanning filters,” *JOSA A*, vol. 10, no. 7, pp. 1499–1508, 1993.
- [15] G. D. Finlayson, and M. S. Drew, “The maximum ignorance assumption with positivity,” in *Color Imaging Conference*, no. 1, pp. 202–205, 1996.
- [16] S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge university press, 2004.
- [17] P. L. Vora and H. J. Trussell, “Mathematical methods for the design of color scanning filters,” *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 312–320, 1997.
- [18] J. R. Magnus, and N. Heinz, *Matrix differential calculus with applications in statistics and econometrics*, John Wiley & Sons, 2nd ed., 2007.
- [19] G. Strang, “The discrete cosine transform,” *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.
- [20] D. P. Bertsekas, “Projected Newton methods for optimization problems with simple constraints,” *SIAM Journal on control and Optimization*, vol. 20, no. 2, pp. 221–246, 1982.
- [21] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk, “What is the space of spectral sensitivity functions for digital color cameras?,” in *IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 168–179, 2013.
- [22] Y. Zhu, “Designing a physically-feasible colour filter to make a camera more colorimetric,” in *London Imaging Meeting*, in press, 2020.
- [23] K. Barnard, L. Martin, B. Funt, and A. Coath, “A data set for color research,” *Color Research & Application*, vol. 27, no. 3, pp. 147–151, 2002.
- [24] G. D. Finlayson, and Y. Zhu, “An improved optimization method for finding a color filter to make a camera more colorimetric,” in *Electronic Imaging: Color Imaging: Displaying, Processing, Hardcopy, and Applications proceedings*, 2020.

Author Biography

Yuteng Zhu is currently a PhD candidate with the Colour and Imaging Lab at the University of East Anglia, UK. She received her dual MSc degree from Zhejiang University, China, and École Centrale de Marseille (ECM), France, in 2014. Her work has mainly focused on filter design for accurate color measurement.

Graham Finlayson is a Professor of Computing Sciences at the University of East Anglia where he leads the Colour & Imaging Lab. Professor Finlayson is interested in ‘computing how we see’ and his research spans computer science (algorithms), engineering (embedded systems) and psychophysics (visual perception). Significantly, some of Graham’s research is implemented and used in commercial products including photo processing software, dedicated image processing hardware and in embedded camera software.