

An Open-Source Algorithm for Metamer Mismatch Bodies

Paul Centore

© February 4, 2017

Abstract

Suppose that two different imaging devices produce two colour images of the same physical object, or that one device images the same object under two different illuminants. A metamer mismatch occurs when two colours are identical in one image, but different in the other image. Each colour output in the first image could have been produced by a large set, called a metamer set, of reflectance spectra. In the second image, the metamer set might produce a wide variety of colour outputs, referred to in their entirety as a metamer mismatch body (MMB). This article presents a simple algorithm for constructing an MMB, by expressing its vertex in an arbitrary direction as a solution to a linear programming program. The algorithm finds not only vertices, but also the reflectance spectra that, when imaged by the second device, produce those vertices. Geometric and algebraic structures behind the algorithm are defined and analyzed in detail. While the linear programming approach to MMBs is not new, the current paper aims to make it accessible, rigorous, and easy to use; to this end, there is an accompanying open-source Octave/Matlab implementation, containing about 30 source lines of code.

Index Terms: metamer, metamer mismatch body, metamer mismatch volume, sensor, illuminants, linear programming, colour

1 Introduction

The problem of metamer mismatching can occur when two different sensing devices image the same object, or when one device images the same object twice, but under different illuminants. Some important practical instances occur when the first “device” is a human visual system, and the second device is a camera; the human observes a scene and then takes a picture of it. There can be a mismatch between the human’s perceptions and the colours in the picture, so some colour correction is necessary. In another instance, a camera photographs the same object twice, outdoors, but at different times of day, under different lighting conditions. The two photographs can produce different colours, making it difficult to determine the object’s local colour. A geometric object called a *metamer mismatch body*^{??} (MMB) describes the extent of the mismatching in such cases. This paper provides a careful

exposition, and open-source implementation, of a standard algorithm^{??-??} for constructing MMBs.

An input for a sensing device, or indeed the human visual system, is a spectral power distribution (SPD) over the visible spectrum, between 400 and 700 nm; as is typical, this paper will discretize the visible spectrum into 31 wavelengths, 10 nm apart. The set of outputs, or responses, that a device can produce is called its *colour space*. A device maps an SPD into a point in the device's own colour space. This article assumes, as is true of many cameras and the human visual system, that the device's colour space is a subset of the vector space \mathbb{R}^3 , and that the mapping is a linear transformation. Typically, a camera's output space is coordinatized by red-green-blue (*RGB*) coordinates, while human colour space uses the *XYZ* coordinates introduced by the Commission Internationale de l'Éclairage (CIE) in 1931.^{??}

An object's local colour is given by a reflectance spectrum $x(\lambda)$, which is a function, taking on values between 0 and 100%, of the 31 typical wavelengths: $x(\lambda)$ is the percentage of incoming light of wavelength λ that the object reflects. Let χ denote the set of all reflectance spectra. χ is a subset of the vector space \mathbb{R}^{31} of real functions on the 31 wavelengths. The device's output for that object depends not only on the device and the object, but also on the illuminant, because the SPD that reaches the device is the product of the object's reflectance spectrum and the illuminant's SPD. Write the illuminant's SPD as a function $p(\lambda)$, which gives a power level at each wavelength.

A device's output is determined by three response curves (called *colour-matching functions* when discussing human vision). Each response curve is a linear functional on the vector space of SPDs. The device's total *response function* can be written as a matrix R of three rows and 31 columns, where each row gives the coefficients for one response curve. The colour output for an object of spectrum $x(\lambda)$ results from two steps. First, the illumination, of SPD $p(\lambda)$, reflects off the object, of spectrum $x(\lambda)$, producing a new SPD, given by wavelength-wise multiplication: $p(\lambda)x(\lambda)$. Second the SPD $p(\lambda)x(\lambda)$ impinges on the sensing device, which responds by producing a colour output $R \cdot (p(\lambda)x(\lambda))$, where \cdot indicates matrix multiplication, and $p(\lambda)x(\lambda)$ is written as a vertical vector of 31 components, one for each wavelength. Combine R and $p(\lambda)$ into one *colour signal map*, given by a matrix

$$\Phi_{ij} = R_{ij}p(\lambda_j), \tag{1}$$

where j indexes the 31 wavelengths. Then Φ is a linear transformation from the 31-dimensional vector space of reflectance spectra to the 3-dimensional device colour space.

In metamer mismatching, two colour signal maps, Φ and Ψ , and one spectrum, x , produce two colour outputs,

$$z = \Phi(x) \text{ and} \tag{2}$$

$$w = \Psi(x), \tag{3}$$

in two colour spaces. Ideally, some function between these spaces would uniquely associate z and w , without requiring any knowledge of x . If we started with the colour output z for an object when imaged by the first device, this function would predict the colour output w for that object when imaged by the second device. In most practical cases, however, there

is no such function. Instead, many colours w_0 in the second colour space are consistent with z , in that there exists some reflectance spectrum x_0 for which

$$z = \Phi(x_0) \text{ and} \tag{4}$$

$$w_0 = \Psi(x_0). \tag{5}$$

The spectrum x_0 is likely different from the original x , but Equations (??) and (??) imply that

$$\Phi(x_0) = \Phi(x). \tag{6}$$

The set of all colours w_0 in the second space, for which an x_0 can be found that satisfies Equations (??) and (??) is called a *metamer mismatch body* (MMB), and denoted $\mathcal{M}(z; \Phi, \Psi)$. $\mathcal{M}(z; \Phi, \Psi)$ is a subset of the colour space of Ψ . The bigger $\mathcal{M}(z; \Phi, \Psi)$ is, the more dissimilar, loosely speaking, Φ and Ψ are; the smaller the mismatch body, the more similar they are. In the limit, the mismatch body is a single point for each z , Φ and Ψ contain identical information, and there exists an invertible linear transformation between their colour spaces.

To find the metamer mismatch body, begin with the *metamer set* of z under Φ , which is the set of all the reflectance spectra that Φ sends to z . It is given formally by

$$\Phi^{-1}(z) \cap \chi, \tag{7}$$

where

$$\Phi^{-1}(z) = \{x \in \mathbb{R}^{31} | \Phi(x) = z\} \tag{8}$$

is the *pre-image* of z . The intersection with χ restricts the pre-image to physically possible reflectance spectra—those whose values are all between 0 and 1. Any element of the metamer set could serve as x_0 in Equations (??) and (??), so the metamer mismatch body, which is the set of all possible w_0 's in Equation (??), is given by

$$\mathcal{M}(z; \Phi, \Psi) = \Psi(\Phi^{-1}(z) \cap \chi). \tag{9}$$

While Equation (??) is straightforward, actually calculating MMBs for practical purposes is surprisingly difficult. Section 3.8.5 of Ref. ?? outlines various algorithms, some using numerical or statistical approximations, and others using linear programming (LP). In 2014, Logvinenko, Funt, and Godau?? (LFG) presented a non-approximate algorithm that did not use linear programming. The current paper presents an explicit, detailed derivation of a previously known LP-based algorithm, along with an open-source Octave/Matlab implementation. The implementation consists of a single routine, containing about 30 source lines of codes, and is ready for immediate use.

In the algorithm in this paper, each direction in the second device's colour space produces a pair of linear programming problems whose solutions are two vertices of \mathcal{M} . At those vertices, \mathcal{M} just touches two parallel supporting planes that are “normal” to the given direction. By stepping through a finely spaced set of directions, the metamer mismatch body can be accurately constructed.

The linear programming formulation depends on the fact that $\Phi^{-1}(z) \cap \chi$ and \mathcal{M} are convex polytopes. The metamer set $\Phi^{-1}(z) \cap \chi$ is the intersection of the affine subspace $\Phi^{-1}(z)$ and the unit cube χ , both of which are convex subsets of \mathbb{R}^{31} . $\Phi^{-1}(z) \cap \chi$ is a polytope, so the intersection is therefore a convex polytope. \mathcal{M} is the linear image under Ψ of the convex polytope $\Phi^{-1}(z) \cap \chi$, and so is itself a convex polytope.

Linear programming optimizes a linear functional over a convex polytope. The linear functional F in this case is the “distance” in a given direction, in the second colour space. F can be pulled back from a functional on \mathbb{R}^3 to a functional Ψ^*F on \mathbb{R}^{31} :

$$(\Psi^*F)(x) = F(\Psi(x)). \tag{10}$$

As a linear functional on \mathbb{R}^{31} , Ψ^*F is also defined over the polytope $\Phi^{-1}(z) \cap \chi$. If a reflectance spectrum x minimizes or maximizes Ψ^*F over $\Phi^{-1}(z) \cap \chi$, then, as the paper will show, the vertex $\Psi(x)$ minimizes or maximizes F over $\mathcal{M}(z; \Phi, \Psi)$. Thus we find not only vertices of \mathcal{M} , but also the spectra that produce those vertices. The geometrical theory of linear programming will show that each such spectrum has value 0 or 1 for at least 28 of the 31 wavelengths; at most three of spectrum values are strictly between 0 and 1.

The paper is organized as follows. After the introduction, the geometric structure of metamer sets and metamer mismatch bodies is derived in detail, leading to a simple algorithm for the construction of MMBs. The algorithm is then summarized in pseudo-code, and an Octave/Matlab implementation, consisting of a single routine with about 30 source lines of code, is presented. An example is given, involving Illuminants A and D65, and the human visual system. The algorithm is then discussed in the context of other algorithms. Finally, a summary of the paper is given.

2 An Algorithm for Metamer Mismatch Bodies

2.1 Metamer Mismatching

Metamer mismatching is an important practical concern, that appears in multiple settings. For one example, most cameras mimic the colour performance of the human visual system: ideally, each *RGB* output of a camera corresponds to exactly one CIE *XYZ* tristimulus vector, and vice versa. In such a situation, a linear transformation could automatically convert a digital image into human perceptual space. In practice, response differences between human cones and artificial photoreceptors, as well as uncertainty about the illumination for the images, prevent a simple correspondence. Instead, a particular *RGB* output could result from many different reflectance spectra. The set of *XYZ* colours those spectra would produce, if viewed by a human, form a metamer mismatch body.

Another instance of metamer mismatching occurs when a camera images an object under one illuminant, producing a certain *RGB*, and one would like to “correct” the image so that the object appears as it would under a different illuminant. This problem is not well-defined, however, because, even if the first illumination is known, many different reflectance spectra could have produced that *RGB*—and those reflectance spectra would produce different colours under the second illuminant. The set of all such possible colours is another instance of a metamer mismatch body.

In its most general setting, a metamer mismatching involves

1. Two sensing devices (one or both of which could be a human visual system),
2. Two illuminants, and
3. One colour output of the first sensing device.

The sensing devices are assumed to be linear and three-dimensional. Mathematically, a device’s colour output, also called a response, is a vector in \mathbb{R}^3 , such as familiar *RGB* or *XYZ* coordinates, and that vector is a linear transformation of a stimulus SPD that impinges on the device. Ordinary function addition and scalar multiplication naturally make the set of SPDs into a vector space. To avoid degeneracy, both illuminants are assumed to be positive everywhere on the visible spectrum; most natural illuminants satisfy this assumption. One further restriction is made: the colour output results from imaging an object, rather than from imaging a light source. The object reflects incoming light in accordance with some reflectance spectrum, which, however, is assumed not to be known.

Suppose, then, that the first device produces a known colour output z , under a known illuminant, when imaging a physical object. Now suppose that a second device images that object, under a second illuminant. A second colour output will be produced, which will depend on the device’s response function, the illuminant, and the object’s reflectance spectrum. While the first two of these factors are known, the reflectance spectrum is not known—in fact, all we know is that it produces the response z when imaged by the first device under the first illuminant. Many reflectance spectra satisfy this condition, and could produce many different responses in the second image. This set of responses, which is a subset of the second device’s colour space, is a metamer mismatch body, for which the following sections will outline a simple, rigorous construction.

2.2 Algebraic Setting

A physical object will be supposed to have a reflectance spectrum $x(\lambda)$, where λ is a wavelength in the visible spectrum. For colour science applications, it is sufficient if λ takes on the 31 wavelengths between 400 and 700 nm, at increments of 10 nm; this paper will follow that practice. The value of x at λ gives the fraction of incoming light at wavelength λ that the object reflects. On physical grounds, each value of $x(\lambda)$ must be between 0 and 1.

An image is produced when some light reflects off the object and enters a sensing device. The output image depends on both the spectral power distribution $p(\lambda)$ of the illumination and the response characteristics of the device. $p(\lambda)$ is modeled as a strictly positive function over the 31 wavelengths; each value is the power level of the illumination at a particular wavelength. The stimulus that the camera responds to is also a spectral power distribution, denoted by $\sigma(\lambda)$ and given by element-wise multiplication:

$$\sigma(\lambda) = p(\lambda)x(\lambda). \tag{11}$$

The output of a particular pixel of the image will be assumed to be a vector of three coordinates, or channels; human vision and most digital cameras satisfy this assumption. The output z_i of the i^{th} channel is given by a linear response curve $R_i(\lambda)$. The coefficient R_{ij} is the value of R_i at the j^{th} wavelength. Using the standard 31 wavelengths for λ , a device’s

output function can be written as a matrix R of 3 rows and 31 columns, populated by the coefficients R_{ij} . If a stimulus $\sigma(\lambda)$ is written as a vertical vector with 31 entries, then the three-channel output z of the sensor is given by

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,31} \\ R_{2,1} & R_{2,2} & \dots & R_{2,31} \\ R_{3,1} & R_{3,2} & \dots & R_{3,31} \end{bmatrix} \begin{bmatrix} \sigma(\lambda_1) \\ \sigma(\lambda_2) \\ \sigma(\lambda_3) \\ \dots \\ \sigma(\lambda_{31}) \end{bmatrix}. \quad (12)$$

The entries z_1 , z_2 , and z_3 are commonly denoted X , Y , and Z when referring to human vision, and R , G , and B when referring to a camera. For the human visual system, the response curves $R_i(\lambda)$ are the familiar CIE colour-matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$.

Equations (??) and (??) allow imaging functions to be expressed as linear transformations between appropriate vector spaces. The first vector space of interest is \mathbb{R}^{31} , which models both stimuli and reflectance spectra. An arbitrary physical stimulus σ can be seen as a non-negative function on the standard 31 wavelengths, and thus as a vector in \mathbb{R}^{31} . A reflectance spectrum can be seen as a function of the standard wavelengths, all of whose values are between 0 and 1. Geometrically, the set χ of reflectance spectra is the solid unit cube in \mathbb{R}^{31} . χ is easily seen to be a convex polytope. The second vector space of interest is \mathbb{R}^3 , where the output vector z lives, and which is the *colour space* for a particular device. Mathematically, a device's colour space can contain any vector in \mathbb{R}^3 , but in practice colour spaces are limited to a small subset.

The matrix in Equation (??) is a linear transformation from \mathbb{R}^{31} to \mathbb{R}^3 , that expresses how a sensing device converts an input stimulus into an output colour signal. A more important transformation, $\Phi : \mathbb{R}^{31} \rightarrow \mathbb{R}^3$, called a *colour signal map*, can be constructed by substituting Equation (??) into Equation (??) and manipulating some terms:

$$z = \begin{bmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,31} \\ R_{2,1} & R_{2,2} & \dots & R_{2,31} \\ R_{3,1} & R_{3,2} & \dots & R_{3,31} \end{bmatrix} \begin{bmatrix} p(\lambda_1)x_1 \\ p(\lambda_2)x_2 \\ p(\lambda_3)x_3 \\ \dots \\ p(\lambda_{31})x_{31} \end{bmatrix} \quad (13)$$

$$= \begin{bmatrix} R_{1,1}p(\lambda_1) & R_{1,2}p(\lambda_2) & \dots & R_{1,31}p(\lambda_{31}) \\ R_{2,1}p(\lambda_1) & R_{2,2}p(\lambda_2) & \dots & R_{2,31}p(\lambda_{31}) \\ R_{3,1}p(\lambda_1) & R_{3,2}p(\lambda_2) & \dots & R_{3,31}p(\lambda_{31}) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{31} \end{bmatrix} \quad (14)$$

$$= \Phi(x). \quad (15)$$

Though physically restricted to the set χ of reflectance spectra, mathematically Φ is defined on all of \mathbb{R}^{31} . The mathematical interpretation will be used later, and the physical restriction will be rewritten as a set of constraints. The transformation Φ describes how a sensing device maps an object colour with reflectance spectrum $x(\lambda)$ to a colour signal, and accounts for both the sensing device's response function and the illumination.

2.3 Metamerism and Metamer Mismatching

Using images to extract information about an object's local colour, which depends only on the object's reflectance spectrum, is an important practical problem, which is surprisingly challenging because one object, when viewed under differing illumination, can produce stimuli with significantly different SPDs. The differing response functions of various sensing devices lead to further complications. The main problem is metamerism, whose meaning varies somewhat with context. In this paper, two different reflectance spectra are said to be *metameric*, or *metamers*, if they produce the same colour signal z under a colour signal map Φ . Algebraically, a colour signal z in \mathbb{R}^3 , could result from any reflectance spectrum in $\Phi^{-1}(z)$, where the exponent -1 indicates the *pre-image* of z :

$$\Phi^{-1}(z) = \{x | \Phi(x) = z\}. \quad (16)$$

This pre-image is called the *metamer set* of z , relative to Φ .

The pre-image of a vector under a linear transformation is an affine subspace, which is always closed and convex, of the domain vector space. The definition in Equation (??) would allow vectors with negative components, or components greater than 1, but reflectance spectra only take on values between 0 and 1, so on physical grounds $\Phi^{-1}(z)$ will be restricted to its intersection with χ . The set

$$\Phi^{-1}(z) \cap \chi \quad (17)$$

is the intersection of two convex sets and is therefore convex. Both sets are also closed, so their intersection is closed. In addition, since χ is compact, its closed subset $\Phi^{-1}(z) \cap \chi$ is therefore also compact.

Now suppose that a second colour signal map, Ψ , which differs from Φ in illuminant or response function (or both), has a second metamer set,

$$\Psi^{-1}(w) = \{x | \Psi(x) = w\}, \quad (18)$$

relative to Ψ . (The w in Equation (??) lives in the colour space of the second device, not the first.) Likely these two metamer sets will be different. *Metamer mismatching* results when two reflectance spectra x and y are metameric under Φ , i.e. they give the same colour signal under Φ , but give different colour signals under Ψ :

$$\Phi(x) = \Phi(y), \quad (19)$$

$$\Psi(x) \neq \Psi(y). \quad (20)$$

Some special cases of metamer mismatching are *illuminant-induced metamer mismatching*, when Φ and Ψ arise from the same sensor but different illuminants, and *observer-induced metamer mismatching*, when Φ and Ψ arise from different sensors but the same illuminant. Mismatching also occurs when both sensors and both illuminants are different, although this case has no common name. The mathematical development in this paper handles all those cases.

To visualize geometrically the extent of metamer mismatching, which varies with colour, define the metamer mismatch body of a colour signal z under two imaging transformations Φ and Ψ by

$$\mathcal{M}(z; \Phi, \Psi) = \Psi(\Phi^{-1}(z) \cap \chi). \quad (21)$$

An MMB can also be associated with a reflectance spectrum x :

$$\mathcal{M}(x; \Phi, \Psi) = \Psi(\Phi^{-1}(\Phi(x)) \cap \chi). \quad (22)$$

If the spectrum x produces the colour signal z under Φ , then Equations (??) and (??) give the same set. Any metamer mismatch body is a subset of the second device's colour space, and so is also a subset of \mathbb{R}^3 . Since $\Phi^{-1}(z) \cap \chi$ was shown to be compact and convex, it follows that $\mathcal{M}(x; \Phi, \Psi)$, which is a linear image of $\Phi^{-1}(z) \cap \chi$, is also compact and convex.

2.4 Constraints for a Metamer Set

Expression (??) writes the pre-image of a colour signal as the intersection of an affine subspace in \mathbb{R}^{31} with the set χ of reflectance spectra, and the accompanying discussion shows that $\Phi^{-1}(z) \cap \chi$ is compact and convex. This section shows in addition that $\Phi^{-1}(z) \cap \chi$ is a polytope, and can be expressed as a finite intersection of closed half-spaces. Writing each closed half-space as a linear constraint will allow a linear programming formulation for a metamer mismatch body.

Let the set of x_i 's, where each x_i is the value of a reflectance spectrum at the i^{th} wavelength, be a basis for \mathbb{R}^{31} . A reflectance spectrum is always between 0 and 1, so

$$0 \leq x_i \leq 1, \quad (23)$$

for $i = 1, 2, \dots, 31$. Linear programming assumes implicitly that all variables are non-negative, so just write

$$x_i \leq 1. \quad (24)$$

These 31 inequality constraints define the set χ of reflectance spectra.

A standard technique in linear programming is *slack variables*, which convert inequalities into equalities. Like ordinary variables, slack variables are required to be non-negative. For each i , introduce the slack variable x_{i+31} by defining

$$x_i + x_{i+31} = 1. \quad (25)$$

Then any reflectance spectrum in χ can be written with 62 variables that satisfy the 31 linear *equality* constraints in Equation (??).

The pre-image $\Phi^{-1}(z)$ can similarly be defined by a set of linear equalities. Writing $z = (z_1, z_2, z_3)$, the condition $z = \Phi(x)$ can be written as three equations, one for each row of Φ :

$$\sum_{i=1}^{31} \Phi_{ij} x_i = z_j, \quad (26)$$

where $j = 1, 2, 3$.

Any reflectance spectrum in $\Phi^{-1}(z) \cap \chi$ must satisfy both the 31 linear constraints in Equation (??) and the three linear constraints in Equation (??). Conversely, any spectrum which satisfies those 34 constraints must be in $\Phi^{-1}(z) \cap \chi$. All the constraints can be

summarized in matrix form:

$$\begin{bmatrix} \Phi & 0_{3 \times 31} \\ I_{31} & I_{31} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{30} \\ x_{31} \\ x_{32} \\ \dots \\ x_{61} \\ x_{62} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}. \quad (27)$$

Denote the 34-by-62 matrix on the left by \mathbf{A} , and the 34-element vector on the right by \mathbf{b} . Refer to the 62-element column vector on the left as \mathbf{x} . Then Equation (27) can be written more succinctly as

$$\mathbf{Ax} = \mathbf{b}. \quad (28)$$

A reflectance spectrum $x(\lambda)$ belongs to $\Phi^{-1}(z) \cap \chi$ if and only if its 31 x_i s, and the 31 slack x_i 's (given by $x_{i+31} = 1 - x_i$, for $i = 1, 2, \dots, 31$) satisfy Equation (28). This constraint form for $\Phi^{-1}(z) \cap \chi$ will be used later in a linear programming formulation.

2.5 The Vertices of a Metamer Set

Since the metamer set $\Phi^{-1}(z) \cap \chi$ can be written as a finite intersection of closed half-spaces (taken from the constraints of the previous section), it must be a convex polytope, and therefore the convex hull of a finite set of *extreme points*, known more familiarly as *vertices*. An extreme point of a convex body is not contained in the (relative) interior of any line segment contained in the body. Roughly speaking, extreme points are the ‘‘corners’’ of the body. This section will apply the geometric theory of linear programming to Equation (28) to find the form of the extreme points of $\Phi^{-1}(z) \cap \chi$.

The canonical linear programming problem^{??} is to maximize a linear functional of the vector \mathbf{x} subject to the constraints

$$\mathbf{Ax} = \mathbf{b}, \quad (29)$$

$$\mathbf{x} \geq 0, \quad (30)$$

where \mathbf{A} has m rows and s columns. (See, for example, Equations (2) and (3) in Section 1.5 of Ref. ??.) Any vector \mathbf{x} that satisfies (29) and (30) is called *feasible*. It can be shown that any linear functional on the polytope defined by (29) and (30) attains its maximum at some extreme point of the polytope, so it is helpful to characterize the extreme points. Theorems 1.8 through 1.11 of the aforementioned reference show that a point is an extreme point if and only if it is a *basic* solution to ?? and ??, where a basic solution is one for which at least $s - m$ variables (where s and m are the dimensions of the matrix \mathbf{A}) are 0. In the current case, \mathbf{A} has 34 rows and 62 columns, so \mathbf{x} is a vertex if and only if at least 28 of its entries are 0.

The first 31 entries of \mathbf{x} are reflectance values for particular wavelengths, while the last 31 are slack variables, each of which pairs with a reflectance value to sum to 1. Since x_i and x_{i+31} (for $i \leq 31$) must sum to 1, no x_i and x_{i+31} can both be 0. Furthermore, if one of them is 0, then the other one must be 1. If neither of the pair is 0, then they must both be strictly between 0 and 1. To satisfy all these conditions, at least 28 of the 31 reflectance values are either 0 or 1; the remaining three can be (but do not have to be) strictly between 0 and 1. These conditions are necessary and sufficient: a reflectance spectrum is a vertex of the metamer set $\Phi^{-1}(z) \cap \chi$ if and only if it satisfies Equations (??) and (??), and at least 28 of its 31 reflectance values are either 0 or 1.

As simple as this criterion is, the high dimension of χ makes calculating all the vertices computationally prohibitive—a total of $2^{28} \binom{31}{3}$, or 1.2 trillion, linear systems?? would have to be solved to check for all possible vertices. (By restricting reflectance spectra to convex combinations of a small set of fairly smooth spectra, which is sufficient for many practical purposes, Finalyson and Morovic?? obtain a simpler metamer set, where solving for all vertices is feasible. This paper, however, considers the more general case.) Fortunately, $\mathcal{M}(z; \Phi, \Psi)$ can be found without finding all the metamer set’s vertices. The next section will use an LP algorithm to quickly find a few extreme points of $\Phi^{-1}(z) \cap \chi$, but only ones which Ψ maps to extreme points of $\mathcal{M}(z, \Phi, \Psi)$.

2.6 The Vertices of a Metamer Mismatch Body

While $\Phi^{-1}(z) \cap \chi$ is a polytope in \mathbb{R}^{31} , the metamer mismatch body $\mathcal{M}(z; \Phi, \Psi)$ is a polytope in \mathbb{R}^3 , a space of much lower dimension. Like a metamer set, a metamer mismatch body is also a convex polytope, because $\mathcal{M}(z; \Phi, \Psi)$ is the linear image under Ψ of the convex polytope $\Phi^{-1}(z) \cap \chi$. Furthermore, every vertex of $\mathcal{M}(z; \Phi, \Psi)$ must be the image of some vertex of $\Phi^{-1}(z) \cap \chi$ —but not vice versa: Ψ will map many vertices of $\Phi^{-1}(z) \cap \chi$ into the interior of $\mathcal{M}(z; \Phi, \Psi)$. If enough vertices of $\mathcal{M}(z; \Phi, \Psi)$ can be identified, then $\mathcal{M}(z; \Phi, \Psi)$ can be well approximated as the convex hull of those vertices; since $\mathcal{M}(z; \Phi, \Psi)$ lives in \mathbb{R}^3 , the number of vertices needed is not prohibitive. This section will calculate vertices of a metamer mismatch body, for a given direction, by formulating and solving a pair of LP problems on the associated metamer set.

In \mathbb{R}^3 , a direction can be formalized using a space-filling stack of parallel planes, each of which is “normal,” in some appropriate inner product, to the vectors in that direction. Alternatively, the planes can be seen as surfaces of constant value for some linear functional $F : \mathbb{R}^3 \rightarrow \mathbb{R}$, given by

$$F(X, Y, Z) = \alpha_X X + \alpha_Y Y + \alpha_Z Z, \tag{31}$$

where X , Y , and Z are a basis for \mathbb{R}^3 . Each plane can be written

$$\alpha_X X + \alpha_Y Y + \alpha_Z Z = \alpha, \tag{32}$$

for some α . The coefficients $(\alpha_X, \alpha_Y, \alpha_Z)$ can be thought of as a direction vector in XYZ -space.

Any compact convex body such as $\mathcal{M}(z; \Phi, \Psi)$ will be intersected by some of the planes resulting from F , and not intersected by others. Since the planes given by Equation (??)

fill \mathbb{R}^3 , there will be two *supporting planes*,^{??} which touch the body only on its boundary, without intersecting the interior. The α 's for the supporting planes will be the minimum and maximum values of F over the convex body. When the body is a convex polytope, at least one of its vertices intersects the minimum plane, and at least one vertex intersects the maximum plane; in the generic case, there is just one minimum and maximum vertex.

The functional F on \mathbb{R}^3 can be pulled back to a functional Ψ^*F on \mathbb{R}^{31} by Ψ :

$$(\Psi^*F)(x) = F(\Psi(x)). \quad (33)$$

The pullback Ψ^*F is a composition of two linear functions, so is itself a linear functional on \mathbb{R}^{31} . In matrix notation,

$$(\Psi^*F) = F(\Psi) \quad (34)$$

$$= \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix} \begin{bmatrix} \Psi_{1,1} & \Psi_{1,2} & \cdots & \Psi_{1,31} \\ \Psi_{2,1} & \Psi_{2,2} & \cdots & \Psi_{2,31} \\ \Psi_{3,1} & \Psi_{3,2} & \cdots & \Psi_{3,31} \end{bmatrix} \quad (35)$$

$$= \begin{bmatrix} \sum_{i=1}^3 \alpha_i \Psi_{i,1} & \sum_{i=1}^3 \alpha_i \Psi_{i,2} & \cdots & \sum_{i=1}^3 \alpha_i \Psi_{i,31} \end{bmatrix}. \quad (36)$$

The pullback is useful because the maximum value of Ψ^*F on $\Phi^{-1}(z) \cap \chi$ is also the maximum value of F on $\Psi(\Phi^{-1}(z) \cap \chi)$. Furthermore, suppose the maximum value of F on $\Psi(\Phi^{-1}(z) \cap \chi)$ occurs at some point w_0 . We will prove by way of contradiction that the maximum value of Ψ^*F on $\Phi^{-1}(z) \cap \chi$ occurs somewhere on $\Psi^{-1}(w_0)$. Suppose not, i.e. suppose that Ψ^*F achieves its maximum on $\Phi^{-1}(z) \cap \chi$ at some point y_0 , such that $\Psi(y_0) \neq w_0$. Since y_0 is a maximum, it must be that

$$F(\Psi(y_0)) > F(\Psi(x_0)), \quad (37)$$

for every x_0 in $\Phi^{-1}(z) \cap \chi$, and in particular for any x_0 such that $\Psi(x_0) = w_0$. Such an x_0 always exists, so

$$F(\Psi(y_0)) > F(w_0). \quad (38)$$

Both w_0 and $\Psi(y_0)$ are in $\mathcal{M}(z; \Phi, \Psi)$, however, so the value of F at $\Psi(y_0)$ exceeds the value of F at w_0 , contradicting the fact that F attains a maximum on $\mathcal{M}(z; \Phi, \Psi)$ at w_0 . Since Ψ^*F attains its maximum on $\Phi^{-1}(z) \cap \chi$ at y_0 , and since any y_0 in the pre-image of w_0 would give the same value of F as w_0 , it must be that

$$F(\Psi(y_0)) = F(w_0), \quad (39)$$

as was to be shown.

Note that Equation (??) does not necessarily imply that $\Psi(y_0) = w_0$. In fact, many points of $\mathcal{M}(z; \Phi, \Psi)$ could give the same value of F as w_0 , if an edge or face of $\mathcal{M}(z; \Phi, \Psi)$ were flush with a level set of F . Then any y_0 that Ψ maps to that edge or face will provide a maximum for Ψ^*F . In the generic case, the maximum of F occurs at a single vertex w_0 of $\mathcal{M}(z; \Phi, \Psi)$, so $\Psi(y_0)$ is a vertex. In fact, the theory of linear programming^{??} proves that at least one vertex of a convex polytope must maximize a linear functional F . Even when $\Psi(y_0)$ is not a vertex, however, it must still be on the boundary of \mathcal{M} .

To find vertices of $\mathcal{M}(z; \Phi, \Psi)$, then, choose a direction $(\alpha_X, \alpha_Y, \alpha_Z)$, which defines a functional F , and find a point y_0 in $\Phi^{-1}(z) \cap \chi$, that maximizes Ψ^*F . The point $\Psi(y_0)$ is then a vertex, or at least a boundary point, of $\mathcal{M}(z; \Phi, \Psi)$.

Since the metamer set $\Phi^{-1}(z) \cap \chi$ can be expressed as the set of solutions to Equations (??) and (??), maximizing or minimizing Ψ^*F on $\Phi^{-1}(z) \cap \chi$ is a standard LP problem:

$$\text{Maximize (or minimize) } (\Psi^*F)\mathbf{x}, \text{ subject to} \quad (40)$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (41)$$

$$\mathbf{x} \geq 0. \quad (42)$$

Equation (??) gives \mathbf{A} and \mathbf{b} , Equation(??) gives Ψ^*F , and any LP package can be used to solve (??) through (??). For this paper, the open-source Octave/Matlab routine `glpk.m` was used. In fact, `glpk.m` can incorporate the box form of the constraints for χ directly, allowing the user to input minimum values of 0 and maximum values of 1 for all the x_i rather than using slack variables. Once slack variables are eliminated, the matrix \mathbf{A} reduces to the matrix Φ , and the vector \mathbf{b} reduces to $[z_1, z_2, z_3]^T$. The call

$$x = \text{glpk}(\Psi^*F, \Phi, [z_1, z_2, z_3]^T, 0_{31 \times 1}, 1_{31 \times 1}) \quad (43)$$

then produces a reflectance spectrum x that minimizes Ψ^*F , and $\Psi(x)$ is a vertex of the metamer mismatch body. Ψ^*F can be maximized by minimizing $-\Psi^*F$. The minimum and maximum vertices are on opposite sides of \mathcal{M} , and a pair of parallel supporting planes can be drawn through them.

2.7 Constructing a Metamer Mismatch Body

Any convex polytope such as a metamer mismatch body can be expressed as the convex hull of its vertices; the open-source Octave/Matlab routine `convhulln.m` makes the calculation quick and easy. The previous section showed how to find two vertices for $\mathcal{M}(z, \Phi, \Psi)$, in a given direction. By choosing a fine enough set of directions, we can generate a set of vertices whose convex hull approximates the metamer mismatch body as closely as desired. This section gives details of such a construction for \mathcal{M} .

Directions in \mathbb{R}^3 can be parametrized by spherical coordinates, which involve a distance r , an azimuth angle θ , and an elevation angle ϕ :

$$\alpha_X = r \cos \theta \sin \phi, \quad (44)$$

$$\alpha_Y = r \sin \theta \sin \phi, \quad (45)$$

$$\alpha_Z = r \cos \phi, \quad (46)$$

where θ goes from 0° to 360° , and ϕ goes from 0° degrees (along the positive Z -axis) to 90° (in the X - Y -plane). Typically, ϕ extends from 0° to 180° , but since two opposite vectors have the same direction for our purposes, it is sufficient to extend ϕ to 90° . Similarly, changing r does not change the direction, so r will be set to 1. With these simplifications, two angles ϕ and θ define a direction.

To find a set of directions in \mathbb{R}^3 that are finely and evenly spaced, define the angle increment

$$\Delta_i = 45^\circ / (2^i), \quad (47)$$

as a function of i , and choose θ s between 0° and 360° , and ϕ s between 0° and 90° , in increments of Δ_i . Each pair in the cross-product of the θ s and the ϕ s defines a direction, whose coefficients as a functional F can be found by substitution into Equations (??) through (??). When i is 5, the angles are spaced at just over a degree apart, which seems to define the metamer mismatch body in adequate though not excessive detail. For finer approximations, of course, larger values of i can be used.

For each pair of angles, solve the LP problem in Equation (??), finding the minimum and maximum vertices of $\mathcal{M}(z; \Phi, \Psi)$ for the direction defined by those angles. The result will be a large collection of vertices. Some vertices will appear multiple times, because there are multiple supporting planes through them; for instance, many supporting planes can be drawn through a corner of a cube. Other times, the same vertex will appear twice, but with very slightly different coordinates, due to numerics. Use the open-source Octave/Matlab routine `convhull.m` to handle both these conditions. Given any set of points in \mathbb{R}^n , this routine finds the vertices of the convex hull of those points, as well as finding the faces of dimension $n - 1$ on the boundary of the convex hull. In \mathbb{R}^3 , `convhull.m` returns a subset of the input vertices, and a set of triangles, each of whose corners is an input vertex. The triangles define the boundary of $\mathcal{M}(z; \Phi, \Psi)$, and can be used in plotting routines or for further analysis.

The interior of the boundary returned by `convhull.m` approximates $\mathcal{M}(z; \Phi, \Psi)$ from below, because the returned vertices are a subset of all the body's vertices. The interior is therefore a subset of $\mathcal{M}(z; \Phi, \Psi)$. A superset of \mathcal{M} can be found by using the supporting planes rather than the vertices. Each supporting plane defines a half-space which contains \mathcal{M} , and any convex body is the intersection of all the half-spaces that contain it.?? A crude, but still informative, superset can be formed from the minimum and maximum planes normal to the X -, Y -, and Z -directions. These planes form a rectangular box that encloses \mathcal{M} , and \mathcal{M} touches each of the box's six sides. Fig. 6 of Ref. ?? is an earlier example of such a box; the current paper extends that approach by finding further minimum and maximum planes in other directions, and thus further refining the approximation. As i increases, the subset and the superset both converge to the MMB, with the superset always containing the subset.

2.8 A Pseudo-Code Summary of the Algorithm

1. The algorithm's inputs are
 - (a) A set of three response curves for a (first) sensing device,
 - (b) An SPD for the illuminant under which that device makes an image,
 - (c) A set of three response curves for a second sensing device (which might be identical to the first sensing device),
 - (d) An SPD for the illuminant under which the second device makes an image, and
 - (e) A three-dimensional vector z , in the colour space of the first device.

2. Calculate the colour signal map Φ (resp. Ψ), which is the composition of the illumination of an object, followed by the first (resp. second) sensor's response. If response functions and illuminants are tabulated at the 31 10-nm intervals between 400 and 700, then Φ and Ψ are both linear transformations from \mathbb{R}^{31} to \mathbb{R}^3 .
3. Find constraints for the metamer set $\Phi^{-1}(z) \cap \chi$, where
 - (a) $\Phi^{-1}(z) = \{x \in \mathbb{R}^{31} | \Phi(x) = z\}$ is the mathematical pre-image of z under Φ , and
 - (b) $\chi = \{x \in \mathbb{R}^{31} | 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, 31\}$ is the subset of \mathbb{R}^{31} consisting of all vectors with components between 0 and 100%. Physically, χ represents the set of reflectance spectra, and $\Phi^{-1}(z) \cap \chi$ is the metamer set of z .

The conditions

$$\Phi(x) = z \tag{48}$$

and

$$0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, 31 \tag{49}$$

together provide a set of constraints that define the metamer set $\Phi^{-1}(z) \cap \chi$.

4. Choose a fineness level i ($i = 5$ or 6 is usually adequate), which gives a corresponding angle increment $45^\circ / (2^i)$. Construct a set of azimuth angles θ , from 0 to 360° , and a set of elevation angles ϕ , from 0 to 90° , at that increment. For each pair (θ, ϕ) , calculate

$$\alpha_X = \cos \theta \sin \phi, \tag{50}$$

$$\alpha_Y = \sin \theta \sin \phi, \tag{51}$$

$$\alpha_Z = \cos \phi, \tag{52}$$

and interpret them as the components of a direction vector in the colour space of the second device.

5. For each direction vector in Step ??,
 - (a) Construct the corresponding linear functional F on \mathbb{R}^3 , given by

$$F(X, Y, Z) = \alpha_X X + \alpha_Y Y + \alpha_Z Z. \tag{53}$$

- (b) Construct the pullback Ψ^*F , which is a linear functional on \mathbb{R}^{31} . Ψ^*F has 31 components, the i^{th} of which is

$$(\Psi^*F)_i = \alpha_X \Psi_{1i} + \alpha_Y \Psi_{2i} + \alpha_Z \Psi_{3i}. \tag{54}$$

- (c) Find a reflectance spectrum $x_{\min} \in \chi$ which minimizes Ψ^*F over the set $\Phi^{-1}(z) \cap \chi$. Formally, this minimization is a standard linear programming problem: minimize the functional Ψ^*F on \mathbb{R}^{31} , subject to the constraints in (??) and (??). In Octave or Matlab, the call

$$x_{\min} = \text{glpk}(\Psi^*F_Y, \Phi, z, 0_{31 \times 1}, 1_{31 \times 1}) \tag{55}$$

performs this minimization.

(d) Similarly, maximize Ψ^*F by minimizing $-\Psi^*F$ with the Octave call

$$x_{\max} = \text{glpk}(-\Psi^*F_Y, \Phi, z, 0_{31 \times 1}, 1_{31 \times 1}). \quad (56)$$

(e) $\Psi(x_{\min})$ and $\Psi(x_{\max})$ are both vertices of $\mathcal{M}(z, \Phi, \Psi)$. Maintain these vertices in a running list.

6. The convex hull of all the vertices is an approximation (from below) to $\mathcal{M}(z, \Phi, \Psi)$.

3 Implementation

`DrawMetamerMismatchBody.m`, an Octave/Matlab implementation of the algorithm just presented, is freely available as an open-source download. The implementation is a single short routine, totaling, apart from input checking and display commands, just under 30 source lines of code. Other researchers are welcome to use and improve this implementation, provided that they make any of their modifications publicly available.

The implementation was investigated with the following example. Suppose the first sensing device was the CIE 1931 Standard Observer, who was viewing an object of a light grey colour, under Illuminant A. Then the response matrix R has 31 columns, and its three rows are the CIE colour-matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$. Denote the power levels for A by $p_A(\lambda)$; scale A so that the the luminance (Y) of ideal white is 100. Then the colour signal matrix Φ for human vision under A is constructed by multiplying each entry of R by the power level of A for that entry:

$$\Phi_{ij} = R_{ij}p_A(\lambda_j). \quad (57)$$

Now suppose that the same object is viewed under Illuminant D65, given by $p_{D65}(\lambda)$, and scaled similarly to A. The second sensing device is again the human eye, so the response matrix R is identical. Like Equation (??), the colour signal matrix Ψ is given by

$$\Psi_{ij} = R_{ij}p_{D65}(\lambda_j). \quad (58)$$

It will be supposed the the object's light grey colour is produced by a flat reflectance spectrum which reflects 50% of the incoming light at any wavelength. Then the coordinates of the object under illuminant A are

$$z = \Phi \cdot [0.5, 0.5, \dots, 0.5]^T \quad (59)$$

$$= [54.8, 50.0, 17.8]^T, \quad (60)$$

which is a vector in human colour space, expressed in CIE XYZ coordinates.

To find the metamer mismatch body $\mathcal{M}(z; \Phi, \Psi)$, find minimum and maximum vertices in any given direction. Let us choose the luminance direction, given by the CIE coordinate Y , with corresponding functional

$$F_Y(X, Y, Z) = \alpha_X X + \alpha_Y Y + \alpha_Z Z \quad (61)$$

$$= 0 \cdot X + 1 \cdot Y + 0 \cdot Z \quad (62)$$

$$= Y. \quad (63)$$

The vector $[0, 1, 0]$ of coefficients for F_Y can be thought of as a direction in XYZ -space. The level surfaces of F_Y are constant-luminance planes that are parallel to the X - Z -plane and fill the entire space. The minimum and maximum planes just touch $\mathcal{M}(z; \Phi, \Psi)$ without intersecting its interior. The body as a whole lies between these planes, and at least one vertex is in each plane.

To minimize or maximize F_Y , which is a functional on three-dimensional colour space, pull it back by Ψ to \mathbb{R}^{31} , producing a linear functional Ψ^*F_Y with 31 components:

$$(\Psi^*F_Y)_i = \alpha_X \Psi_{1i} + \alpha_Y \Psi_{2i} + \alpha_Z \Psi_{3i} \quad (64)$$

$$= 0 \cdot \Psi_{1i} + 1 \cdot \Psi_{2i} + 0 \cdot \Psi_{3i} \quad (65)$$

$$= \Psi_{2i}. \quad (66)$$

Ψ^*F_Y in this case is just the second row of Ψ , because of F_Y 's simple component expression.

Linear programming will minimize and maximize Ψ^*F_Y over $\Phi^{-1}(z_0) \cap \chi$, which is the metamer set of z , i.e. the set of reflectance spectra that are metameric for humans to the 50% reflectance spectrum under Illuminant A. The Octave/Matlab routine `glpk.m` is a convenient LP solver, because the constraints for individual reflectances x_i have the simple box form $0 \leq x_i \leq 1$. The Octave call

$$x_{\min} = \text{glpk}(\Psi^*F_Y, \Phi, [54.8, 50.0, 17.8]^T, 0_{31 \times 1}, 1_{31 \times 1}) \quad (67)$$

finds a reflectance spectrum $x_{\min}(\lambda)$ that minimizes F_Y over $\mathcal{M}(z; \Phi, \Psi)$, while the call

$$x_{\max} = \text{glpk}(-\Psi^*F_Y, \Phi, [54.8, 50.0, 17.8]^T, 0_{31 \times 1}, 1_{31 \times 1}) \quad (68)$$

finds a reflectance spectrum $x_{\max}(\lambda)$ that maximizes it.

Figures ?? and ?? show the minimizing and maximizing reflectance spectra. As must be the case, at least 28 of the percentages are either 0 or 1. In these two cases, exactly 28 are either 0 or 1. The three percentages that are strictly between 0 and 1 are marked with a star. The corresponding vertices of $\mathcal{M}(z; \Phi, \Psi)$ are

$$\Psi \cdot x_{\min} = [54.7, 44.7, 65.0]^T, \quad (69)$$

$$\Psi \cdot x_{\max} = [40.2, 55.3, 43.7]^T, \quad (70)$$

so the planes $Y = 44.7$ and $Y = 55.3$ just bound $\mathcal{M}(z; \Phi, \Psi)$, and touch it at those vertices. For comparison, the Ψ -image of the 50% grey that produced z is

$$\Psi \cdot [0.5, 0.5, \dots, 0.5]^T = [47.5, 50.0, 54.4]^T. \quad (71)$$

It is inside $\mathcal{M}(z; \Phi, \Psi)$, and its Y -coordinate is between 44.7 and 55.3, the minimum and maximum values for Y .

Similarly to F_Y , we could define linear functionals F_X and F_Z that minimize and maximize X and Z over $\mathcal{M}(z; \Phi, \Psi)$. Linear programming will produce planes of the form $X = \dots$ and $Z = \dots$ that bound $\mathcal{M}(z; \Phi, \Psi)$ and touch it at (at least) one vertex. Together, these six planes (a minimum and maximum plane for each of X , Y , and Z) form a rectangular box which is a crude approximation to $\mathcal{M}(z; \Phi, \Psi)$, because $\mathcal{M}(z; \Phi, \Psi)$ is completely inside this

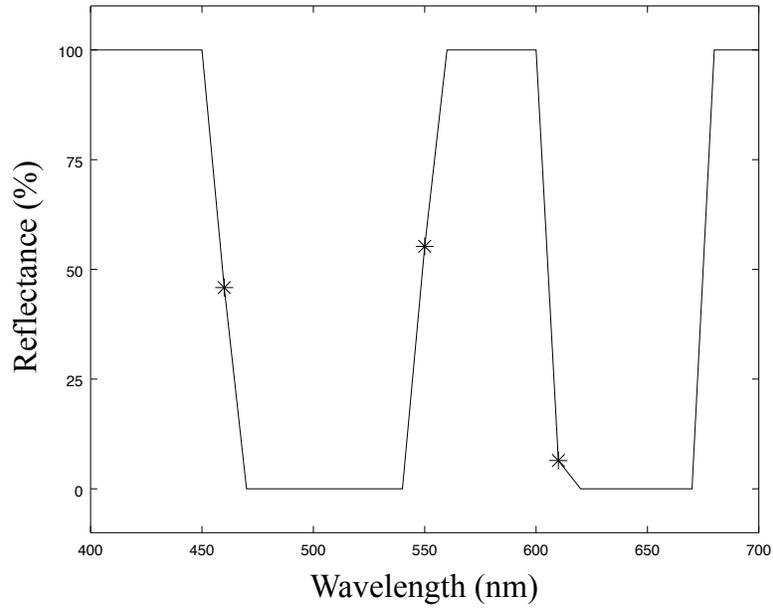


Figure 1: The Minimizing Reflectance Spectrum x_{\min}

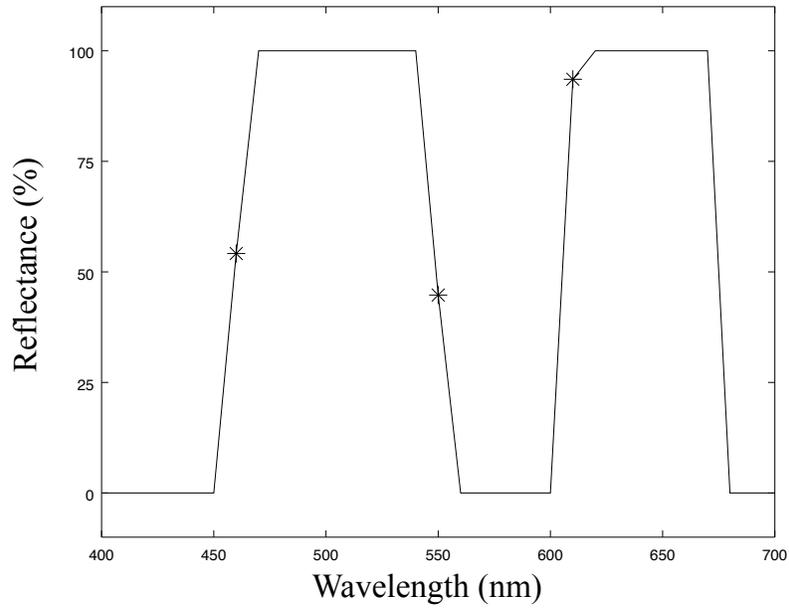


Figure 2: The Maximizing Reflectance Spectrum x_{\max}

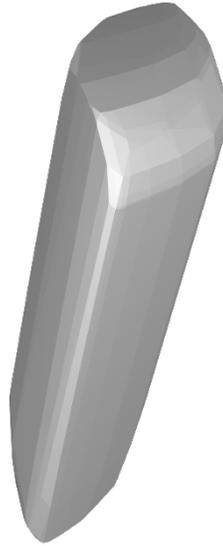


Figure 3: The Metamer Mismatch Body for a 50% Grey, Under Illuminants A and D65

box, and touches each of its six sides. (Ohta and Wyszecki present a similar box in Figure 6 of a 1975 paper.^{??}) Additional bounding planes, say in the direction of the functional $X + Y + Z$, would slice off corner sections of the rectangular box, leading to a polyhedron that better approximates $\mathcal{M}(z; \Phi, \Psi)$ and touches it at more points. Further directions, of course, would further refine the polyhedron.

While a containing polyhedron is one natural approximation, another, perhaps more natural, approximation is the convex hull of the vertices where $\mathcal{M}(z; \Phi, \Psi)$ touches the polyhedron. Figure ?? shows $\mathcal{M}([54.8, 50.0, 17.8]^T, \Phi, \Psi)$ as a convex hull of the vertices obtained by setting $i = 5$ in Equation (??). This value of i produced 3120 different directions and 338 vertices. Increasing i to 6 produced 7224 directions, but only 440 vertices. Since more than doubling the number of directions only increased the number of vertices by about 30%, it is likely that many vertices were being found again, but for different bounding planes. These diminishing returns suggest that, after some point, further refinements will not justify the additional computation.

For the example just presented, the implementation in `DrawMetamerMismatchBody.m` took about 10 seconds when run with $i = 5$, and about 27 seconds when run with $i = 6$. No attempt was made to optimize the implementation, nor to tune the testing computer (a Macbook Pro laptop) for faster performance.

4 Comparison with Previous Work

Section 3.8.5 of Wyszecki and Stiles' classic reference^{??} outlines some algorithms for finding metamer mismatch bodies, including some LP methods. The non-LP methods, however, are generally inexact and slow, and some make debatable (though plausible) assumptions about the density of metamers within the body. Many of the methods generate a large set

of points within the MMB, and fit an ellipsoid to that set, to approximate the boundary—the possibilities for misleading results are obvious. Computationally, the main difficulty is that reflectance spectra have many dimensions, typically 31 for practical applications; consequently, a metamer set cannot be estimated very well, and the same goes for an MMB, which is the linear image of a metamer set.

Another issue is whether an algorithm finds points interior to the MMB, or only on its boundary. Since MMBs are convex, it is determined by its boundary. Non-LP methods typically find many points inside the MMB, and extrapolate out to the boundary. The LFG algorithm and the current algorithm, by contrast, only find boundary points, much reducing the computational load.

Wyszecki and Stiles also discuss LP algorithms. Eugene Allen introduced LP for MMBs in a 1969 paper^{??} and showed some results, although he gave next to no details about the method. In 1975, Ohta and Wyszecki^{??} presented an explicit LP formulation for Allen’s method. While the current paper views an MMB as the convex hull of its vertices, Allen viewed an MMB in terms of its projection onto the X - Y -plane in CIE XYZ space. For a given (X, Y) in the X - Y -plane, an MMB attains a minimum and a maximum Z , which can be joined to make a vertical line segment over (X, Y) . By convexity, the MMB is the totality of these line segments. In our terminology, Allen would have introduced two additional constraints:

$$\sum_{i=1}^{31} \Psi_{1,i} x_i = X, \tag{72}$$

$$\sum_{i=1}^{31} \Psi_{2,i} x_i = Y, \tag{73}$$

and minimized and maximized the functional

$$F_Z = Z. \tag{74}$$

The resulting reflectance spectra have up to five values strictly between 0 and 1, while ours have no more than three—this difference occurs because our reflectance spectra map only to vertices of the MMB, while Allen’s map to arbitrary boundary points. Allen’s method also requires an additional check that there exists at least one spectrum x satisfying Equations (??) and (??). Many current LP implementations, however, automatically check for this kind of consistency.

The current description of the LP approach makes no claim to originality, but rather aims at simplicity, accessibility, and completeness. In 2016, when ready-made software for linear programming and convex hulls is readily available, implementation is much easier than it was in Allen and Wyszecki’s day. The openness, readability and readiness for use of the accompanying routine `DrawMetamerMismatchBody.m` will hopefully save other researchers time and effort. In addition, previous papers have many gaps. For example, Ohta and Wyszecki state without proof (3rd paragraph of p. 329 of Ref. ??) that most reflectance values for spectra of MMB boundary points are either 0 or 1. Schmitt^{??} makes similar statements on p. 603 of his 1976 paper, again without support. While their statements are correct, they should be proven. Sect. ?? of the current paper provides a proof, while other sections fill similar gaps in previous derivations.

The LFG algorithm, developed in 2014, is notable for finding exact boundary points of an MMB, but without using linear programming. Their approach combines Φ and Ψ into a six-dimensional linear transformation Υ , so that both colour signal maps are considered simultaneously. (Allen suggested the same “six-dimensional hypervolume in the combined color space” on p. 38 of Ref. ??.) Points on the boundary of $\mathcal{M}(z; \Phi, \Psi)$ are then related to points on the boundary of $\Upsilon(\chi)$, which, in analogy to Schrödinger’s optimal colours,^{??} are well-approximated by reflectance spectra whose values are either 0 or 1, with at most five transitions between those values. After choosing a starting point within the MMB, and a direction, a set of six equations involving both Φ and Ψ is derived, which finds the boundary point on the MMB in the given direction from the starting point, as well as the corresponding reflectance spectrum. A Matlab implementation then solves those equations numerically, in as many directions as desired.

5 Summary

This paper has presented a simple algorithm, and a concise, open-source Octave/Matlab implementation, for constructing metamer mismatch bodies. A metamer mismatch body is viewed as the convex hull of a set of vertices, and the vertex in a particular direction is formulated as a linear programming problem that minimizes or maximizes a functional in that direction. A metamer mismatch body is the linear image of a metamer set. Since the metamer set is the intersection of an affine subspace (the pre-image of a colour output under a known colour signal map) and the set of reflectance spectra, it can be readily expressed as a set of constraints. The linear functional is pulled back from the mismatch body to the metamer set, and the linear programming problem is solved on the metamer set. The result is a reflectance spectrum, whose image is a vertex of the mismatch body. Once enough vertices in enough directions are known, the convex hull of the vertices is a good representation of the body. The algorithm is derived rigorously, summarized in pseudo-code, and tested with a simple example involving human vision, Illuminant A, and Illuminant D65. A final discussion places the linear programming approach in historical context, showing that it was well-established by the 1970s, but has recently been overlooked. An accompanying Octave/Matlab implementation, `DrawMetamerMismatchBoundary.m`, containing about 30 source lines of code, has been released as open source; other researchers are invited to use and modify it, on the understanding that they will make their modifications publicly available.

References

1. G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed., John Wiley & Sons, 1982.
2. Eugene Allen, “Some New Advances in the Study of Metamerism: Theoretical Limits of Metamerism; An Index of Metamerism for Observer Differences,” *Color Engineering*, Vol. 7, No. 1, pp. 35-40, January-February 1969.
3. N. Ohta & G. Wyszecki, “Theoretical chromaticity-mismatch limits of metamers viewed under different illuminants,” *Journal of the Optical Society of America*, Vol. 65, No. 3,

pp. 327-333, March 1975.

4. F. J. M. Schmitt, "A method for the treatment of metamerism in colorimetry," *Journal of the Optical Society of America*, Vol. 66, No. 6, pp. 601-608, June 1976.
5. Deane B. Judd, "The 1931 I. C. I. Standard Observer and Coordinate System for Colorimetry," *JOSA*, Vol. 23, pp. 359-374, October 1933.
6. Alexander D. Logvinenko, Brian Funt, & Christoph Godau, "Metamer Mismatching," *IEEE Transactions on Image Processing*, Vol. 23, No. 1, pp. 34-43, January 2014.
7. Bernard Kolman & Robert E. Beck, *Elementary Linear Programming with Applications*, 2nd ed., Academic Press, 1995.
8. Graham D. Finlayson & Peter Morovic, "Metamer Sets," *Journal of the Optical Society of America A*, Vol. 22, No. 5, pp. 810-819, May 2005.
9. S. R. Lay, *Convex Sets and Their Applications*, Dover Publications, 2007.