Metamer Constrained Colour Correction

G. D. Finlayson, P. M. Morovič
School of Information Systems
University of East Anglia,
Norwich, UK

Abstract

At an early stage in almost all colour reproduction pipelines device RGBs are transformed to CIE XYZs. This transformation is called colour correction. Because the XYZ matching functions are not a linear combination of device spectral sensitivities there are some colours which look the same to a device but have quite different XYZ tristimuli. That such device metamerism exists is well known, yet the problem has not been adequately addressed in the colour correction literature. In this paper, we examine in detail the role that metamers play in developing a new colour correction algorithm.

Our approach works in two stages. First, for a given RGB we characterise these to fall possible camera metamers. In the second stage this set is projected onto the XYZ colour matching functions. This results in a set of XYZs any one of which might be the correct answer for colour correction. Good colour correction results by choosing the middle of the set. We call the process of computing the set of metamers, projecting them to XYZs and performing selection, *metamer constrained colour correction*.

Experiments demonstrate that our new method significantly outperforms traditional linear correction methods. For the particular case of saturated colours (these are among the most difficult to deal with) the error is halved on average; the maximum error is reduced by a factor of 4.

1. Introduction

The problem of *colour correction* is the problem of mapping RGB sensor responses to CIE XYZ tristimulus values. Unless device sensitivities are a linear transform of the XYZ colour matching functions (the Luther conditions apply) then perfect colour correction is not possible. Formally, there exist pairs of colours that look the same to a device, they are device metamers, but they have different XYZs (and vice versa).

Perhaps the simplest way to deal with device metamerism, and that used by most colour correction algorithms, is to use regression. Here RGBs are mapped to corresponding XYZs so that some error criterion is minimised (usually root mean square error). Regression is a pragmatic approach but it does not explicitly address the metamerism problem. In carrying out colour correction we are really trying to predict the XYZ that would be induced by the reflectance that induced the RGB. No statement is made about reflectance in regression, yet we would like to make such a statement since reflectance is at the heart of the colour correction problem.

In contrast, in lighting matrix colour correction, an algebraic as opposed to regression formulation, the role that reflectance plays is made clear. The central assumption in lighting matrix color correction is that surface reflectance can be modelled by a linear sum of three basis functions: $S(\lambda) = \sigma_1 S(\lambda) + \sigma_2 S_2(\lambda) + \sigma_3 S_3(\lambda)$ (where i indexes basis function and σ_i is the ith weighting coefficient). Relative to this assumption, it is straightforward to show that RGBs and reflectance coefficients (the σ weights) are a 3×3 camera "lighting" matrix transform apart. XYZs are also linearly related to surface weights by an XYZ lighting matrix. By pre-multiplying an RGB by the inverted camera lighting matrix, surface weights are recovered. By multiplying the recovered sigma weights by the XYZ lighting matrix we effectively solve the color correction problem.

Unfortunately lighting matrix correction works no better than least-squares. In fact it can never work any better since a regression solution must be optimal (with respect to the error that is minimized). However, the lighting matrix formulation is helpful because it makes the role of reflectance explicit and this turns out to be very useful when we think about how colour correction might be improved. Specifically, because the approach works by attempting to recover reflectance we make a statement, for an individual RGB, about whether we think the recovery is "plausible" or "implausible". If, for example, the recovered reflectance has negative reflectance values or reflects more than 100% of incident light then it is implausible (such reflectances are physically impossible) and so we can predict poor correction performance in this case.

In this paper, we extend and improve the lighting matrix correction method by enforcing "physical realizeability" constraints. To do this we must move from a 3-dimensional model of reflectance to one that has more degrees of freedom. While this will allow us to get round physical implausibility it also reintroduces device metamerism: there is more than one physically realizeable reflectance corresponding to each RGB. To address this problem we, at the first stage in colour correction, characterise the entire set of RGB metamers. In the second stage this set is projected down on to the XYZ matching curves to give a set of candidate XYZs (for colour correction). Of course to complete colour correction a single answer must be chosen from this set. We propose that the centroid, or "middle" of the candidate set is an good choice to make since it mitigates against the worst case error. The process of solving for the set of RGB metamers, then characterizing the corresponding set of XYZs and finally choosing a single representative member of that set is called *Metamer Constrained Colour Correction*.

Experiments demonstrate that our new metamer constrained correction method significantly outperforms least-squares correction. It delivers much lower correction error. The greatest performance increase is for saturated colours (the colours where conventional colour correction works least well) where the mean error is reduced by a factor of 2 and the maximum error diminishes by a factor of 4.

Importantly, metamer constrained colour correction is a very simple procedure to implement. The physical realizeability constraints can be formulated as linear inequalities. Moreover, solving for the metamer constrained set of XYZs amounts to maximizing and minimizing a small set of linear objective functions. It follows then that metamer constrained colour correction involves solving a small set of linear programs for each RGB. Linear programming is an extremely fast computational procedure.

In section 2 we review the linear least-squares and linear lighting matrix correction methods. Metamer constrained correction is presented in section 3. Various experiments are reported in section 4. The paper finishes with a short conclusion in section 5.

2. Linear Colour Correction

The easiest and most straightforward method (e.g. see Horn [3]) for mapping RGB to XYZ is to use a *linear transformation* in the form of a 3×3 matrix $\mathbf{M}$ satisfying:

$$\mathbf{X} = \mathbf{MR} \tag{1}$$

where $\mathbf{X}$ is a $3 \times n$ matrix of XYZ tristimulus values (under a standard illuminant) and $\mathbf{R}$ is a $3 \times n$ matrix of RGB sensor responses. Having a set of such responses and their corresponding tristimulus values, one can solve for $\mathbf{M}$ in the *least squares* sense, minimizing the root mean squared error:

$$||\mathbf{X} - \mathbf{MR}||^{\mathbf{2}} \tag{2}$$

This approach is guaranteed to deliver good results in two cases only [9, 1]. First if the sensor sensitivities of the RGB sensors are a linear transformation of the XYZ colour-matching functions, second if the reflectance data used is three-dimensional – that is, if the matrix $\mathbf{M}$ is used to transform data from a locally linear area of colour space. Typically camera sensors are not linearly related to XYZs. Nor are reflectances 3-dimensional. So, the error in (2) is non-zero.

The linear transformation matrix $\mathbf{M}$, will of course, depend on the data-set used to obtain it. There is therefore a trade-off when using least squares: colours which appear frequently (in the training set) are corrected well, those that appear less frequently are corrected less well. Because there are more colours clustered around the achromatic axis than there are at the extremes of the object colour solid, desaturated colours tend to be corrected with much less color error than saturated colours. This said, the challenge for colour correction is to reduce the error in correcting the saturated colours without affecting the very good colour correction performance which is generally delivered for desaturated colours.

In thinking about colour correction, and how it might be improved, it is imperative to understand how RGBs and XYZs are formed. The real goal of colour correction is to find the reflectance that induced an RGB and then to calculate the XYZ for this reflectance. This apparently simple insight is the basis for a second linear correction method: lighting matrix colour correction[5].

A lighting matrix is a 3x3 matrix which is a function of device spectral sensitivities, illumination and, surface reflectance. Let $\mathbf{e}$ denote the 31×1 column vector[1] of the illuminant, D an operator making a diagonal matrix out of a column vector, $\mathbf{R}$ the 31×3 matrix containing the RGB sensor sensitivities of a set of sensors and $\mathbf{B}$ the $31 \times n$ matrix: an n-dimensional set of surface reflectance basis functions, then the lighting matrix (which is $n \times 3$) is defined as:

$$\mathbf{\Lambda^e} = \mathbf{B^T D(e) R} \tag{3}$$

The role of this matrix becomes clear when considering how reflectances[2] relate to RGBs. Let us denote by σ the $n \times 1$ column vector of the weights, and by ρ the 3×1 sensor response, then:

$$\sigma^{\mathbf{T}} \mathbf{B^T D(e) R} = \sigma^{\mathbf{T}} \mathbf{\Lambda^e} = \rho^{\mathbf{T}} \tag{4}$$

The lighting matrix informs us that the RGB is an $n \times 3$ linear transform from the n dimensional surface weight vector. If n is 3, then the weights σ can be recovered from RGB using a simple matrix inverse operation:

$$\sigma^{\mathbf{T}} = \rho^{\mathbf{T}} (\mathbf{\Lambda^e})^{-\mathbf{1}} \tag{5}$$

[1] Following convention, spectra are represented here by their values at 31 sample points across the visible spectrum (400 nm to 700 nm in 10 nm intervals).

[2] When linear models are used to represent reflectances, then each particular reflectance is defined by the weights for these basis functions – the final reflectance being a weighted sum of the basis functions.

If we denote the lighting matrix for the XYZ colour matching functions by $(\mathbf{\Lambda^e_X})$then an RGB vector ρ can be mapped to the XYZ tristimulus $\mathbf{x}$:

$$\mathbf{x^T} \;=\; \rho^{\mathbf{T}}(\mathbf{\Lambda^e})^{-\mathbf{1}}(\mathbf{\Lambda^e_X}) \tag{6}$$

The mapping in (6) is also a simple 3×3 matrix. However, the role that reflectance plays is explicitly modelled.

3. Metamer constrained colour correction

A three-dimensional linear model actually fits a lot of reflectances rather well: especially whites and greys and desaturated colours. As colours become more saturated so the model becomes less accurate. Intuitively, this is to be expected, desaturated reflectances are very smooth and so are composed mostly of low frequency components. Saturated colours tend to have much higher frequency components (e.g. a deep red has almost 0 reflectance in the blue part of the spectrum and this can shoot up to 70 or 80% in the longer wavelengths). The 3-dimensional linear model is insufficiently rich to model higher frequencies. This failure manifests itself in inaccurate and implausible reflectance recovery. The recovered reflectance for a saturated colour often has reflectance values that are bigger than 100% or less than 0% (they reflect or absorb more light than was incident).

In order to model saturated colours and so facilitate accurate correction higher dimensional models of reflectance are needed. However, given an n-dimensional model ($n > 3$) of surface reflectance, the system of equations defined by the lighting matrix becomes under-determined. Instead of a single unique solution a whole set of solutions becomes feasible. The set of solutions of such a system can be expressed as [6]:

$$\rho^{\mathbf{T}} = \sigma^{\mathbf{T}}\mathbf{\Lambda^e} = \sigma_\rho^{\mathbf{T}}\mathbf{\Lambda^e} + \sigma_{\mathbf{0}}^{\mathbf{T}}\mathbf{\Lambda^e} \tag{7}$$

$$\sigma_\rho^{\mathbf{T}}\mathbf{\Lambda^e} = \rho^{\mathbf{T}} \tag{8}$$

$$\sigma_{\mathbf{0}}^{\mathbf{T}}\mathbf{\Lambda^e} = \mathbf{0} \tag{9}$$

Here $\sigma_{\mathbf{0}}$ is a set of weights characterising reflectances which account for zero RGB response, they are *black* for the sensor. The final reflectance is then represented as a sum of a reflectance which gives the required response (σ_ρ) and a sequence of *metameric black* reflectances ($\sigma_{\mathbf{0}}$) all giving zero response to the camera.

In Equations (7)-(9) we describe reflectances by that part which projects non-trivially on to RGB and that part which is orthogonal or black to the camera. Any n-dimensional basis set can be split into two parts such that the first 3 basis vectors project non-trivially onto the sensors and the second $n-3$ vectors are Metameric blacks. Henceforth we will assume our reflectance basis is in this form:

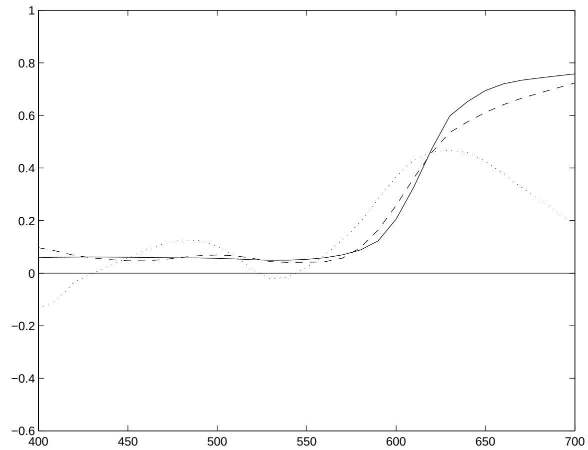

Figure 1: An example of 5 dimensional recovery.

$$\begin{aligned} &[\sigma_{\mathbf{1}}\ \sigma_{\mathbf{2}}\ \sigma_{\mathbf{3}}\ \mathbf{0}\ \cdots\ \mathbf{0}]_\rho^{\mathbf{T}}\mathbf{\Lambda^e} = \rho^{\mathbf{T}} \\ &[\mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \sigma_{\mathbf{4}}, \sigma_{\mathbf{5}}\ \ldots \sigma_{\mathbf{n}}]^{\mathbf{T}}\mathbf{\Lambda^e} = \mathbf{0} \end{aligned} \tag{10}$$

3.1. Feasibility Constraints

Not all reflectance functions described in eq.(7) are feasible. Surface reflectance functions must be *non-negative* (no less than no light is reflected by a surface) and *less than or equal to one* (no more than all light is reflected by a surface)[3]. These conditions restrict the the σ_i parameters. The feasible set consists only of reflectances must satisfy these plausibility constraints.

Figure 1 shows reflectance recovery subject to constraints. The two horizontal solid lines, $y = 1$ and $y = 0$, denote the area of feasible solutions (non-negative and less than or equal to one), the solid line shows the original reflectance, the dotted line shows the 3D lighting matrix solution (which in this case has negatives so is not even feasible). The dashed line shows the five dimensional solution obtained from the 3D solution and the metameric black solution. Notice that as well as being feasible the 3D+metameric black solution is also much closer to the actual reflectance.

That reflectances have between 0 and 100% reflectance is just one constraint that might usefully be applied. We can also place constraints on the σ weights themselves. It could be for example that σ_4, the coefficient controlling the contribution of the 4th basis function, must lie in the interval $[-0.02, 0.02]$. In order to understand how

[3]This is not true for all reflectances (e.g. for *fluorescent* reflectances) these however are disregarded here.

these intervals are chosen we must understand how the reflectance basis functions themselves are derived. Suppose that $\mathbf{U}$ denotes a $31 \times m$ matrix of representative surface reflectances; each column of $\mathbf{U}$ contains a single surface reflectance. We would like to find a $31 \times n$ basis ($n << m$) $\mathbf{B}$ such that linear combinations of the columns of $\mathbf{B}$ could be used to approximate $\mathbf{U}$. The technique of characteristic vector analysis[4] allows us to find such a basis. Associated with $\mathbf{B}$ we have an $n \times m$ weight matrix $\mathbf{W}$ such that:

$$\mathbf{BW} \approx \mathbf{U} \tag{11}$$

where $\mathbf{W}$ is chosen to minimize the approximation error (actually it is defined by a least-squares regression matrix). The minimum and the maximum of all weight sets for each of the basis functions, that is the minimum and maximum of the rows of $\mathbf{W}$, serve as the lower and upper bounds for the σ weights[4].

With all these constraints in hand we next show how we can use them in colour correction. First we observe that each constraint can be written as an inequality e.g. the reflectance must be less than or equal to 1. Second, we note that there will probably be many reflectances that satisfy all the inequalities and so we need to choose one answer from the set. In making this choice it is reasonable to suppose that we wish to optimize some error criterion. Assuming the error criterion is linear then metamer constrained colour constancy can be formulated as a linear program.

3.2. Linear Programming

Linear Programming is defined by a set of inequalities (half-spaces) and a linear objective function which is to be minimised (or maximised), formally:

$$\min_{\sigma} \mathbf{c^T} \sigma \tag{12}$$

$$\text{subject to } \mathbf{A}\sigma \leq \mathbf{b} \tag{13}$$

where $\mathbf{A}$ is a $k \times n$ matrix of the left side of the inequalities, σ is a $n \times 1$ column vector of the unknown reflectance, $\mathbf{b}$ is a $k \times 1$ column vector of the right sides of the inequalities and $\mathbf{c}$ is a $n \times 1$ column vector defining the objective function. The reader is reminded that n is the dimension of surface reflectance.

The constraints for reflectances addressed above can be interpreted as two inequalities for each wavelength (each reflectance at each wavelength *must* be non-negative and less than or equal to one). Because we are representing reflectances by their values at 31 sample points this gives 62 constraints. Added to this we need constraints on the sigma weights. Assuming reflectance is n-dimensional we need two constraints for each basis function and so need $2n$ additional constraints. The $k = 62 + 2n$ constraints together are combined in the constraint matrix $\mathbf{A}$ in eq.(13). Note that the formulation given in (13) allows only "less than or equal to" inequalities but that we want "bigger than or equal to inequalities" as well. The latter inequalities are readily transformed to the former by multiplying the appropriate row of $\mathbf{A}\sigma$ and $\mathbf{b}$ by -1.

We would like to find the set of XYZs that satisfy all the constraints in $\mathbf{A}$. That is, we would like to find all reflectances, characterized by a weight vector σ, and project these down onto the XYZ colour matching functions. To find the set in XYZ space, the objective function (defined by the vector $\mathbf{c}$) was chosen to minimise (as well as maximise) each of the X, Y, and Z co-ordinates in turn. Specifically, $\mathbf{c}$ is one column of the lighting matrix for the XYZ functions (each column of which defines the X, Y, and Z responses to each of the n basis functions). The result of this optimization is six extreme XYZ co-ordinates.

3.3. Finding the Centre of Feasible Cube

Let us consider the set of all possible XYZs (corresponding to an RGB) to be the cube enclosing the six extreme XYZs. This cube[5] is a larger estimate of the solution set, as not all points within the cube necessarily represent a feasible solution.

Selecting a single answer from the cube is straightforward: we simply choose the cube centre. This selection minimizes the cost of making an error in either the X, Y or Z coordinates. It mitigates against the worst case correction error. Though, we point out that the center of the cube need not be feasible. In spite of this, this algorithm (further referred to as the LPCC model *Linear Programming Centre of the Cube*) performs rather well. Moreover, because it is designed to minimize the cost of making a worst case error it should result in small maximum errors. Experiments reported later show that this is the case.

3.4. Centroid of Feasible Set

The advantage of the feasible cube approach is its simplicity. However, we would like to characterise more accurately the feasible solution set. Since, in so doing we should have a stronger foundation for carrying out colour correction.

To to this, We sample the interior points of the cube and for each point check to see if it is feasible (that they satisfy the constraints in 13). Proceeding in this way we

[4] The maxima and minima are picked for each weight dimension separately, therefore the resulting bounds are not "real" in the sense that such a set of weights does not necessarily exist.

[5] The cube is constructed so as to cover the extreme XYZs, and so that all faces are parallel to the planes defining the co-ordinate system.

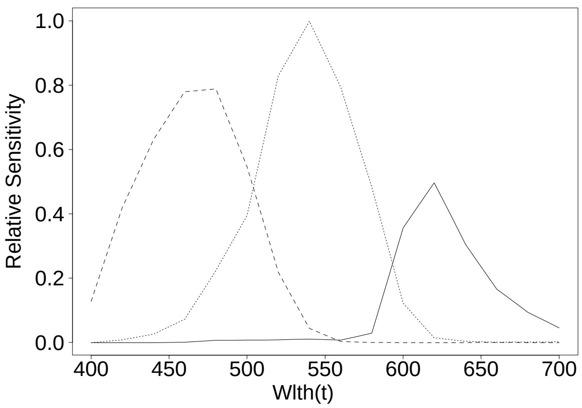

Figure 2: The camera sensitivities used in the experiments.

find a convex cloud of feasible points inside LPCC cube. Relative to this cloud, the point that has smallest maximal distance from all other points in the cloud is the XYZ chosen for correction. This model will be referred to as the LPFSC model (*Linear Programming Feasible Set Centre*).

4. Results

To compare the performance of all the algorithms that have been described, several simulation experiments were carried out. Figure 2 shows the spectral sensitivities of the camera which were used throughout the testing[2]. In a first experiment two standard reflectance sets were used: the Macbeth Color Checker Chart and Munsell colour atlas. In a second experiment, and in order to examine more closely the question of saturated reflectances we selected a set of 134 reflectances which lie close to the boundary of the object colour solid. These reflectances were drawn from four combined reflectance sets: the Munsells, a set of object reflectances [8], a set of natural reflectances [7] together with a set of Dupont dye reflectances. In order to get good correction performance for this saturated set we trained our algorithm on a set of reflectances with as large a gamut as possible. Thus, we selected 41 reflectances which, when projected down to the XYZ colour matching functions span the complete range of colours. Qualitatively, the 41 reflectances are similar to Macbeth reflectances but they led to better algorithm performance for all algorithms tested. The illuminants used for the experiments were CIE standard illuminants D65 and A and a measured fluorescent illuminant.

Correction results are summarised in Tables 1 through 3. Table 1 reports correction CIE Lab error for correcting Macbeth colour checker RGBs to XYZs. The LSQ correction involves fitting RGBs to XYZs. The 3D correction answer is based on a 3-dimensional characteristic vector analysis (CVA) of the Macbeth's. LPCC and LPSFC both use a 6 dimensional CVA of the colour checker. It is clear that the constrained correction methods LPCC and LPFSC perform significantly better; especially, in terms of the maximum error. However, the incorporation of constraints has also reduced the mean error by about 1 delta E. For completeness we show the delta Es for each of the 24 patches on the colour checker. It is apparent that the constrained approach can deliver higher error than LSQ. But, this is as we might expect since the colour correction problem is ill posed. For individual reflectances the LSQ answer may be better than the unconstrained answer.

Table 2 reports results for training on the Macbeth colour checker and testing on the Munsells. Overall the performance trends are as before. The constrained regression delivers a significantly reduced maximum error rate and reduces the mean by about 1 delta E.

Perhaps the most interesting experimental results are reported in Table 3. Here we train on the 41 maximum gamut reflectance set and test on the 134 saturated reflectances (reflectances that are close to the boundary of the object colour solid). We know that the problem area for colour correction is the saturated colours and so we might expect the constrained correction method to work best here. This is the case. The maximum error is reduced by a factor of 4 compared to the 3D and LSQ correction methods. The mean error is reduced by a factor of 2.

The reader will see that the LPFSC method returns slightly higher error rates than LPCC. This was unexpected. The LPCC method models the feasible solutions to the colour correction as a cube of XYZs. However, only some interior points of the cube are actually possible so one might imagine that the LPCC method is suboptimal in some sense. In contrast the LPSFC method works only with the feasible interior points and so should provide a better basis for colour correction. However, the difficulty here is finding the interior points. Our algorithm works by partitioning the interior of the cube and checking feasibility on a point by point basis. If the actual feasible set is very flat (i.e. not very 3-dimensional) then it is possible that we fail to adequately characterise the feasible set and this can lead to poorer correction performance. We are currently developing methods to deal with this problem.

5. Conclusions

A set of colour correction algorithms was presented, two of which are considered to be standard (LSQ and 3D) and two of which are new *metamer constrained colour correction* algorithms (LPCC and LPFSC). We believe the

model	**LSQ**	**3D**	**LPCC**	**LPFSC**
01	1.6629	1.5371	0.8242	1.1306
02	5.8518	5.7990	5.3350	5.3350
03	4.1728	4.0279	5.7413	5.7413
04	4.5731	4.8079	1.5372	2.2052
05	0.6844	0.5349	1.3913	1.3913
06	8.6639	9.0111	3.3010	2.8917
07	4.4621	4.7646	5.5743	5.6261
08	6.9284	6.8586	6.2854	6.0979
09	3.6974	3.5421	2.9690	2.0814
10	4.6480	4.5023	3.1649	3.6381
11	3.2845	3.6841	3.7096	4.1462
12	8.4166	8.7749	7.4726	6.8340
13	12.1185	12.0785	3.2444	3.3862
14	7.3733	7.9298	6.7304	6.9141
15	11.9203	11.6026	4.4801	4.7297
16	0.8639	1.3331	1.4672	3.3274
17	1.7569	1.3288	1.8074	2.2077
18	12.7408	13.0432	4.7075	4.0180
19	2.0171	1.8989	1.9951	2.0862
20	2.2340	2.1320	2.9743	2.9743
21	2.1263	2.0405	3.2006	3.2006
22	1.4463	1.3753	3.1613	3.1613
23	1.2370	1.1878	3.9807	3.8521
24	0.6889	0.6584	2.8924	2.0325
max	12.7408	13.0432	7.4726	6.9141
min	0.6844	0.5349	0.8242	1.1306
mean	4.7321	4.7689	3.6645	3.7087

Table 1: Statistics (ΔE values) for the following set-up: illuminant: D65, training set: Macbeth ColorCheckerChart (24 reflectances), testing set: Macbeth ColourChecker Chart (24 reflectances), dimension: 6 (covering 99.8 % variation)

model	**LSQ**	**3D**	**LPCC**	**LPFSC**
max	20.4060	32.3414	16.3173	15.5799
min	0.1977	0.0688	0.4390	0.1961
mean	5.8408	6.0043	4.9052	4.8511

Table 2: The statistics (ΔE values) for the following set-up: illuminant: D65, training set: uniformly distributed 41 reflectances, testing set: 462 Munsell chips, dimension: 6 (covering 99.3 % variation)

model	**LSQ**	**3D**	**LPCC**	**LPFSC**
max	37.4189	43.5308	11.7975	12.95
min	0.7934	0.7346	0.4448	0.49
mean	8.4883	10.6349	4.4246	4.63

Table 3: The statistics (ΔE values) for the following set-up: illuminant: fluorescent, training set: 41 uniformly distributed reflectances, testing set: 134 saturated reflectances, dimension: 6 (covering 99.3 % variation)

latter improve on the former because they are based on a better conceptual understanding of the problem itself. Specifically, the linear correction methods, LSQ and 3D, assume that colour correction is a 1 to 1 problem. It is not. Rather there is an intrinsic uncertainty in the correction. Many metamers project down onto the same RGB. Yet this metamer set projects non-uniquely on to XYZ. The constrained metamer approach, of which the LPCC and LPFSC algorithms are examples, characterise the feasible set of XYZs and provide a means for selecting a single answer from within the set.

Importantly our new well founded algorithms for colour correction deliver improved correction performance. In all cases error rates are reduced. For the particular case of saturated colours (the colours that are most difficult to correct) the mean and maximum error rate are reduced respectively by a factor of 2 and 4 respectively.

References

[1] M.S. Drew and B.V. Funt. Natural metamers. *CVGIP:Image Understanding*, 1991. In press.

[2] G.D. Finlayson and S.D. Hordley. Recovering device sensitivities using quadratic programming. In *IS&T and SID's 6th Color Imaging Conference*, November 1998.

[3] B.K.P. Horn. Exact reproduction of colored images. *Comp. Vision, Graphics and Image Proc.*, 26:135–167, 1984.

[4] L.T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A*, 3:1673–1683, 1986.

[5] L.T. Maloney and B.A. Wandell. Color constancy: a method for recovering surface spectral reflectance. *J. Opt. Soc. Am. A*, 3:29–33, 1986.

[6] T. Katriňák, M. Gavalec, E. Gedeonová, and J. Smítal. *Algebra a Teoretická Aritmetika*. Alfa Bratislava (Czechoslovakia), 1985.

[7] J. P. S. Parkkinen, J. Hallikanen, and T. Jaaskelainen. Characteristic spectra of munsell colors. *Journal of the Optical Society of America A*, 6:318–322, 1989.

[8] M.J. Vrhel, R. Gershon, and L.S. Iwan. Measurement and analysis of object reflectance spectra. *Color Research and Application*, 19:4–9, 1994.

[9] B.A. Wandell and J.E. Farrell. Watermaloney into wine: Converting scanner RGB to tristimulus XYZ. In *Device-Independent Color Imaging and Imaging Systems Integration*, pages 92–101. SPIE Vol. 1909, 1993.