Metamer constrained colour correction

Graham D. Finlayson & Peter M. Morović

School of Information Systems
University of East Anglia
Norwich NR4 7TJ
United Kingdom

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 colour 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 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 the 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, and the maximum reduced by a factor of four.

I 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 (i.e. the Luther conditions apply) then perfect colour correction is not possible. Formally, there exist pairs of colours that look the same to a device, the device metamers, but which 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.

However, in lighting matrix colour correction\textsuperscript{1,2} an algebraic as opposed to regression formulation, the role that reflectance plays is made clear. The central assumption in lighting matrix colour correction is that surface reflectance can be modelled by a linear sum of three basis functions: $S(\lambda) = \sigma_1 S_1(\lambda) + \sigma_2 S_2(\lambda) + \sigma_3 S_3(\lambda)$ (where $i$ indexes the basis function, $\sigma_i$ is the $i$th weighting coefficient and $S_i(\lambda)$ is the $i$th basis function). Relative to this assumption, it is straightforward to show that RGBs and reflectance coefficients (the $\sigma$ weights) are a $3 \times 3$ 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 colour 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 minimised). However, the lighting matrix formulation is useful because it makes the role of reflectance explicit and in so doing helps us understand the limitations of linear colour correction. Specifically, because the lighting matrix approach works by reflectance recovery we can look at the recovered spectrum and say something about whether it is “plausible” or “implausible”; if, 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 realizability” 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: that there is more than one physically realizable 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 a 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 \textit{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 realizability 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 3.

In section II we review the linear least-squares and linear lighting matrix correction methods. Metamer constrained correction is presented in section III. Various experiments are reported in section IV. The paper finishes with a short conclusion in section V.

II LINEAR COLOUR CORRECTION

The easiest and most straightforward method (e.g. see Horn 4) for mapping RGB to XYZ is to use a linear transformation in the form of a $3 \times 3$ matrix $\mathbf{M}$ satisfying:

$$\mathbf{X} = \mathbf{M} \mathbf{R}$$  \hspace{1cm} (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 (preferably under the same illuminant). 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:

$$||\mathbf{X} - \mathbf{M} \mathbf{R}||^2$$  \hspace{1cm} (2)

The linear transform matrix can be then derived using the Moore-Penrose inverse (pseudo-inverse) 5 of matrix $\mathbf{R}$:

$$\mathbf{M} = \mathbf{X} \mathbf{R}^T (\mathbf{R} \mathbf{R}^T)^{-1}$$  \hspace{1cm} (3)

This approach is guaranteed to deliver good results in two cases only 6,7. First if the sensor sensitivities of the RGB sensors are a linear transformation of the XYZ colour-matching functions (the Luther condition), 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, therefore the error in eq.(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 (see Fig.1), desaturated colours tend to be corrected with much less 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. To address this question we
must first understand how the physical quantities of light and reflectance combine
and interact with the sensor (camera or matching function). Let \( e(\lambda) \), \( s(\lambda) \)
and \( r(\lambda) \) denote the spectral power distribution of illumination, spectral reflectance
and sensor sensitivity. These functions combine together to form a sensor response
according to the following integral equation:

\[
\rho_r = \int_0^\infty s(\lambda) e(\lambda) r(\lambda) d\lambda
\]

(4)

Following convention, we represent spectra by their values at 31 sample points
across the visible spectrum (400 nm to 700 nm in 10 nm intervals). Under this
assumption the integral in eq.(4) can be written as a summation and this summation
can then be written as a matrix equation:

\[
\rho_r = \sum_{i=1}^{31} s_i e_i r_i
\]

(5)

\[
\rho_r = s^T D(e) r
\]

(6)

where \( s \), \( e \) and \( r \) are 31 vectors representing surface reflectance, light and sensor
sensitivity. The function \( D() \) maps the elements in a 31-vector to the diagonal
elements in a 31 \( \times \) 31 diagonal matrix. This step is necessary if we are to remove
the explicit summation from (5).

Let us now try to understand the colour correction problem. We begin by writing
expressions for camera RGB and corresponding XYZ:

\[
\rho = s^T D(e) R
\]

(7)

\[
\chi = s^T D(e) X
\]

(8)

In eq.(7,8) \( R \) and \( X \) are the 31 \( \times \) 3 matrices of RGB sensor sensitivities and XYZ
colour-matching functions respectively and \( \rho \) and \( \chi \) are the 1 \( \times \) 3 RGB and XYZ
responses. In the colour correction problem, both \( s \) and \( \chi \) are unknown.

Clearly there exists a whole set of \( s_i \) which might induce the same RGB. To
see this suppose that \( s^T D(e) R = \rho \). Because \( D(e) R \) is a 31 \( \times \) 3 matrix (only 3
of 31 degrees of freedom are fixed) there must be many vectors \( b_i \) that have the
property that \( b_i^T D(e) R = 0 \): they are metameric blacks. Using the laws of matrix
algebra it follows that \( (s + b_i)^T D(e) R = \rho \). However, because \( D(e) R \) is usually

\[
\rho_r = \int_0^\infty s(\lambda) e(\lambda) r(\lambda) d\lambda
\]
quite different from \(D(e)X\), \(s^TD(e)X\) need not equal \((s + \mathbf{b}_1)^TD(e)X\). That is metamers for a given camera need not look the same to a human observer.

It follows that given no restrictions on camera metamers the colour correction problem is ill-posed: there does not, and cannot, exist a one to one mapping from RGB to XYZ. Drew and Funt \(^9\) side-stepped this ill-posedness by assuming that all surface reflectance spectra could be written as a linear sum of three basis functions\(^1\):

\[
s = \mathbf{B}\sigma\tag{9}
\]

where \(\mathbf{B}\) is a 31x3 basis matrix. Each column of \(\mathbf{B}\) represents a single reflectance basis function. The lighting matrix for a camera records the RGB response of the camera to each of the surface reflectance basis functions:

\[
\Lambda_R^e = \mathbf{B}^TD(e)\mathbf{R}\tag{10}
\]

This simple \(3 \times 3\) matrix simplifies colour correction since it provides an explicit, invertible, statement of how reflectance relates to RGB. To see this, substitute \(s = \mathbf{B}\sigma\) in eq.(7):

\[
\sigma^T\mathbf{B}^TD(e)\mathbf{R} = \sigma^T\Lambda_R^e = \rho\tag{11}
\]

The lighting matrix informs us that the RGB is an \(3 \times 3\) linear transform from the 3 dimensional surface weight vector. It follows, that the weights \(\sigma\) can be recovered from RGB using a simple matrix inverse operation:

\[
\sigma^T = \rho(\Lambda_R^e)^{-1}\tag{12}
\]

Now, if we denote the lighting matrix for the XYZ colour matching functions by \((\Lambda_X^e)\) then an RGB vector \(\rho\) can be mapped to the XYZ tristimulus \(\chi\) vector as follows:

\[
\chi = \rho(\Lambda_R^e)^{-1}\Lambda_X^e\tag{13}
\]

The mapping in (13) is also a simple \(3 \times 3\) matrix. The form of the equation is quite important since it breaks colour correction into two steps. First a reflectance is recovered from an RGB and then the corresponding XYZ is calculated for that reflectance. The idea of reflectance recovery and reprojection is fundamental to our new colour correction method developed in the next section.

\(^1\) In the strictest sense Funt and Drew assumed that all colour signals (light multiplied by reflectance) were modelled by a 3-d linear model. However, for our purposes adopting a 3-d reflectance model allows us to replicate their result.
III METAMER CONSTRAINED COLOUR CORRECTION

Statistical studies\(^9\text{-}^{12}\) have shown that a three-dimensional linear model fits a lot of typical reflectances quite well; especially whites and greys and desaturated colours. For these surfaces, lighting matrix colour correction works quite well. Unfortunately, as colours become more saturated so a 3D model becomes less accurate and as a consequence colour correction performance also drops. Intuitively, we expect this sort of performance profile. Desaturated reflectances are generally quite smooth and so are composed mostly of low frequency components. In contrast 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. Indeed, when saturated colours are described using a 3-d (smooth) reflectance model one often finds that the recovered reflectance for a saturated colour often has 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 \((\Lambda^e_R\) becomes an \(n \times 3\) matrix rather than a \(3 \times 3\) matrix). Instead of a single, unique solution a whole set of solutions become feasible. The set of solutions of such a system can be expressed as\(^{13}\):

\[
\rho = \sigma^T \Lambda^e_R
\]

\[
\sigma = \sigma_\rho + \sigma_0
\]

\[
\sigma_\rho^T \Lambda^e_R = \rho
\]

\[
\sigma_0^T \Lambda^e_R = 0
\]

One way to solve for the particular solution \(\sigma_\rho\) is to solve for it in the least squares sense:

\[
\rho[\Lambda^e_R]^T \Lambda^e_R\]^{-1} [\Lambda^e_R]^T \Lambda^e_R = \rho
\]

\[
\downarrow
\]

\[
\rho[\Lambda^e_R]^T \Lambda^e_R\]^{-1} [\Lambda^e_R]^T = \sigma_\rho^T
\]

\[
\downarrow
\]

\[
\rho[\Lambda^e_R]^+ = \sigma_\rho^T
\]

where \([\Lambda^e_R]^T \Lambda^e_R\]^{-1} [\Lambda^e_R]^T is the pseudo-inverse denoted by \([\Lambda^e_R]^+\). That is, we choose \(\sigma_\rho\) to be a linear combination of the 3 columns of \(\Lambda^e_R\). However, if \(\Lambda^e_R\) is \(n \times 3\) then the columns of \(\Lambda^e_R\) clearly only span a 3-dimensional subspace of \(n - 3\) space and this means that there are \(n - 3\) degrees of freedom left to explore. Let us define an \(n \times (n - 3)\) matrix \(\Lambda^0\) such that:
\[ [\Lambda^0]^T \Lambda^e_R = 0 \quad \& \quad \det([\Lambda^0]^T[\Lambda^0]) \neq 0 \] (19)

where 0 denotes a \( n - 3 \times 3 \) zero matrix, and \( \det() \) is the determinant function. The first condition in (19) tells us that the space spanned by the columns of \( \Lambda^0 \) is orthogonal to the space spanned by the columns of \( \Lambda^e_R \) and therefore that \( \Lambda^0 \) is a matrix describing the solutions to eq.(17) (i.e. the columns of \( \Lambda^0 \) are \( \sigma_0 \)-s). The second condition implies that the columns of \( \Lambda^0 \) are linearly independent.

To find \( \Lambda^0 \) we first need the projector of a \( n \times k \) \( (n \geq k) \) matrix:

\[
P(A) = A [A^T A]^{-1} A^T
\] (20)

From the definition of the projector it can be shown that:

\[
P(\Lambda^e_R) \Lambda^e_R = \Lambda^e_R
\] (21)

\[
P(\Lambda^e_R) \Lambda^0 = 0
\] (22)

\[
(I - P(\Lambda^e_R)) \Lambda^0 = \Lambda^0
\] (23)

\[
P(\Lambda^0) = I - P(\Lambda^e_R)
\] (24)

Equation (24) is quite important as it tells us we can define the projector of \( \Lambda^0 \) (the matrix we are trying to solve for) in terms of the projector of \( \Lambda^e_R \). It is clear then that the columns of \( P(\Lambda^0) \) span the same space as the columns of \( \Lambda^0 \). However, \( P(\Lambda^0) \) is an \( n \times n \) matrix and so some of the columns must be linearly dependent (i.e. some columns can be written as weighted sums of other columns). Thus, to define \( \Lambda^0 \) we must find \( n - 3 \) linearly independent vectors in the column space of \( P(\Lambda^0) \). This might be done in numerous ways. One simple way is to systematically select all sets of \( n - 3 \) columns from \( P(\Lambda^0) \) and check the determinant condition in (19). If the determinant is non-zero the selected columns must be linearly independent and \( \Lambda^0 \) is defined.

In almost all imaginable cases the first \( n - 3 \) columns of \( P(\Lambda^0) \) will be linearly independent and so this combinatorial search procedure turns out to be quite efficient.

Returning to eq.17, \( \sigma_0 \) can be defined as:

\[
\sigma_0 = \Lambda^0 \phi
\] (25)

where \( \phi \) is an arbitrary \( n - 3 \) vector.

**A Feasibility Constraints**

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) \(^2\). These conditions restrict the \( \sigma_i \) coefficients:

\(^2\) This is not true for all reflectances (e.g. for *fluorescent* reflectances) these however are not considered here.
\[0 \leq B\sigma \leq \mathbf{1}. \quad (26)\]

That reflectances have between 0 and 100\% reflectance is just one constraint that might usefully be applied. We can also place constraints on the \( \sigma \) weights themselves. It could be for example that \( \sigma_4 \), the coefficient controlling the contribution of the 4th basis function, must lie in the interval \([-0.02, 0.02]\). In order to understand how 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\(^{10,12}\) allows us to find such a basis. Associated with \( \mathbf{B} \) we have an \( n \times m \) weight matrix \( \mathbf{W} \) such that:

\[
\mathbf{BW} \approx \mathbf{U}
\]

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 \( \sigma \) weights\(^3\), they shall be denoted \( \theta_{\min} \) and \( \theta_{\max} \) (both being \( n \)-dimensional vectors) respectively. So \( \sigma \) must satisfy:

\[
\begin{align*}
\theta_{1\min} \leq \sigma_1 & \leq \theta_{1\max} \\
\theta_{2\min} \leq \sigma_2 & \leq \theta_{2\max} \\
& \vdots \\
\theta_{n\min} \leq \sigma_n & \leq \theta_{n\max}
\end{align*}
\]

(28) \hspace{1cm} (29) \hspace{1cm} (30)

With all these constraints on the \( \sigma \) weights in hand we must consider how they might be used in colour correction. Clearly, we are going to wish to minimize or maximize something. So long as that something is chosen to be a linear function we can use the technique of linear programming.

### B Linear Programming

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

\[
\begin{align*}
\min_{\sigma} & \mathbf{c}^T \sigma \\
\text{subject to} & \ \mathbf{A}\sigma \leq \mathbf{b}
\end{align*}
\]

(31) \hspace{1cm} (32)

\(^3\) 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.
where $A$ is a $k \times n$ matrix of the left side of the inequalities, $\sigma$ is a $n \times 1$ column vector of the unknown weights defining the reflectance, $b$ is a $k \times 1$ column vector of the right sides of the inequalities and $c$ 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 inequalities. Added to this we need constraints on the sigma weights. Assuming reflectance is $n$-dimensional we need two constraints (the lower and upper bound) for each basis function and so need $2n$ additional constraints. The $k = 62 + 2n$ constraints together are combined in the constraint matrix $A$ in eq.(32). Note that the formulation given in eq.(32) 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 $A$ and $b$ by -1.

We would like to find the set of XYZs that satisfy all the constraints in $A$. That is, we would like to find all reflectances, characterized by a weight vector $\sigma$, and project these down onto the XYZ colour matching functions. To find the set in XYZ space, the objective function (defined by the vector $c$) was chosen to minimise (as well as maximise) each of the X, Y, and Z co-ordinates in turn. Specifically, $c$ 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 are six extreme XYZ co-ordinates.

\section{Centre of 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\textsuperscript{4} 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 (though experiments show that in almost all cases it is feasible). In spite of this, this algorithm (further referred to as the LPCC model \textit{Linear Programming Cube Centre}) 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. The advantage of the feasible cube approach is its simplicity. However, we would like to characterize more accurately the feasible solution set. Since, in so doing we should have a stronger foundation for carrying out colour correction.

\textsuperscript{4} 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.
D Centre of the feasible set.

To do this, we sample the interior points of the cube and for each point check to see if it is feasible. Proceeding in this way we find a convex cloud of feasible points inside the LPCC cube. Relative to this cloud, the XYZ that has smallest maximal distance from all other XYZs in the cloud is the answer chosen for correction. This model will be referred to as the LPFSC model (*Linear Programming Feasible Set Centre*).

IV 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.

In a first experiment two standard reflectance sets were used: the Macbeth Colour 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 $^{12}$, a set of object reflectances $^{2}$, a set of natural reflectances together with a set of Dupont dye reflectances (see Fig. 3 for a chromaticity plot of these data sets). 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 (see [http://www.sys.uea.ac.uk/People/graham/](http://www.sys.uea.ac.uk/People/graham/)). 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 illuminant D65 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 LPFSC 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 $\Delta E$s 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.

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 methods 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 XYZ. 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.

V 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 the new *metamer constrained colour correction* algorithms (LPCC and LPFSC). We believe the 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, which 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.

VI ACKNOWLEDGEMENTS

The authors would like to thank Dr. Steven Hordley and Dr. Ján Morovič for their support and useful comments throughout the research. This work was carried
out while the authors were at the Colour & Imaging Institute at the University of Derby. We would like to express our gratitude to that institute for its support of this work.

The research presented in this paper is funded by the Nuffield Foundation, and their support is acknowledged here. The studentship of Peter Morović is funded by Hewlett Packard, which is greatly appreciated.

REFERENCES

FIGURE 1. Metamer distribution frequency gaussian fit plot in xy chromaticity (the vertical line at the peak of the distribution denotes the achromatic point).

FIGURE 2. Camera sensors used for experiments.
FIGURE 3. CIE xy chromaticity plot of data sets used in experiments (for CIE illuminant D65).
### TABLE 1. Statistics ($\Delta E$ values) for the following set-up: illuminant: D65, training set: Macbeth ColourChecker Chart (24 reflectances), testing set: Macbeth ColourChecker Chart (24 reflectances), dimension: 6 (covering 99.8% variation)

<table>
<thead>
<tr>
<th>model</th>
<th>LSQ</th>
<th>3D</th>
<th>LPCC</th>
<th>LPFSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>1.6629</td>
<td>1.5371</td>
<td>0.8242</td>
<td>1.1506</td>
</tr>
<tr>
<td>02</td>
<td>5.8518</td>
<td>5.7990</td>
<td>5.3350</td>
<td>5.3350</td>
</tr>
<tr>
<td>03</td>
<td>4.1728</td>
<td>4.0279</td>
<td>5.7413</td>
<td>5.7413</td>
</tr>
<tr>
<td>04</td>
<td>4.5731</td>
<td>4.8079</td>
<td>1.5372</td>
<td>2.2052</td>
</tr>
<tr>
<td>05</td>
<td>0.6844</td>
<td>0.5349</td>
<td>1.3913</td>
<td>1.3913</td>
</tr>
<tr>
<td>06</td>
<td>8.6639</td>
<td>9.0111</td>
<td>3.3010</td>
<td>2.8917</td>
</tr>
<tr>
<td>07</td>
<td>4.4621</td>
<td>4.7646</td>
<td>5.5743</td>
<td>5.6261</td>
</tr>
<tr>
<td>08</td>
<td>6.9284</td>
<td>6.8586</td>
<td>6.2854</td>
<td>6.0979</td>
</tr>
<tr>
<td>09</td>
<td>3.6974</td>
<td>3.5421</td>
<td>2.9690</td>
<td>2.0814</td>
</tr>
<tr>
<td>10</td>
<td>4.6480</td>
<td>4.5023</td>
<td>3.1649</td>
<td>3.6381</td>
</tr>
<tr>
<td>11</td>
<td>3.2845</td>
<td>3.6841</td>
<td>3.7096</td>
<td>4.1462</td>
</tr>
<tr>
<td>12</td>
<td>8.4166</td>
<td>8.7749</td>
<td>7.4726</td>
<td>6.8340</td>
</tr>
<tr>
<td>13</td>
<td>12.1185</td>
<td>12.0785</td>
<td>3.2444</td>
<td>3.3862</td>
</tr>
<tr>
<td>14</td>
<td>7.3733</td>
<td>7.9298</td>
<td>6.7304</td>
<td>6.9141</td>
</tr>
<tr>
<td>15</td>
<td>11.9203</td>
<td>11.6026</td>
<td>4.4801</td>
<td>4.7297</td>
</tr>
<tr>
<td>16</td>
<td>0.8639</td>
<td>1.3331</td>
<td>1.4672</td>
<td>3.3274</td>
</tr>
<tr>
<td>17</td>
<td>1.7569</td>
<td>1.3288</td>
<td>1.8074</td>
<td>2.2077</td>
</tr>
<tr>
<td>18</td>
<td>12.7408</td>
<td>13.0432</td>
<td>4.7075</td>
<td>4.0180</td>
</tr>
<tr>
<td>19</td>
<td>2.0171</td>
<td>1.8089</td>
<td>1.9951</td>
<td>2.0862</td>
</tr>
<tr>
<td>20</td>
<td>2.2340</td>
<td>2.1320</td>
<td>2.9743</td>
<td>2.9743</td>
</tr>
<tr>
<td>21</td>
<td>2.1263</td>
<td>2.0405</td>
<td>3.2006</td>
<td>3.2006</td>
</tr>
<tr>
<td>22</td>
<td>1.4463</td>
<td>1.3753</td>
<td>3.1613</td>
<td>3.1613</td>
</tr>
<tr>
<td>23</td>
<td>1.2370</td>
<td>1.1878</td>
<td>3.9867</td>
<td>3.8521</td>
</tr>
<tr>
<td>24</td>
<td>0.6889</td>
<td>0.6584</td>
<td>2.8924</td>
<td>2.0325</td>
</tr>
<tr>
<td>max</td>
<td>12.7408</td>
<td>13.0432</td>
<td>7.4726</td>
<td>6.9141</td>
</tr>
<tr>
<td>min</td>
<td>0.6844</td>
<td>0.5349</td>
<td>0.8242</td>
<td>1.1306</td>
</tr>
<tr>
<td>mean</td>
<td>4.7321</td>
<td>4.7689</td>
<td>3.6645</td>
<td>3.7087</td>
</tr>
</tbody>
</table>

### TABLE 2. The statistics ($\Delta E$ 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)

<table>
<thead>
<tr>
<th>model</th>
<th>LSQ</th>
<th>3D</th>
<th>LPCC</th>
<th>LPFSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>20.4060</td>
<td>32.3414</td>
<td>16.3173</td>
<td>15.5799</td>
</tr>
<tr>
<td>min</td>
<td>0.1977</td>
<td>0.0688</td>
<td>0.4390</td>
<td>0.1961</td>
</tr>
<tr>
<td>mean</td>
<td>5.8408</td>
<td>6.0043</td>
<td>4.9052</td>
<td>4.8511</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>model</th>
<th>LSQ</th>
<th>3D</th>
<th>LPCC</th>
<th>LPFSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>37.4189</td>
<td>43.5308</td>
<td>11.7975</td>
<td>12.95</td>
</tr>
<tr>
<td>min</td>
<td>0.7934</td>
<td>0.7346</td>
<td>0.4448</td>
<td>0.49</td>
</tr>
<tr>
<td>mean</td>
<td>8.4883</td>
<td>10.6349</td>
<td>4.4246</td>
<td>4.63</td>
</tr>
</tbody>
</table>