https://doi.org/10.2352/CIC.2022.30.1.31

An exposure invariant neural network for colour correction

***Abdullah Kucuk*[1]*, Graham D. Finlayson*[1]*, Rafal Mantiuk*[2]*, Maliha Ashraf*[3]**

[1]*School of Computing Sciences, University of East Anglia, United Kingdom,*
[2]*Department of Computer Science and Technology, University of Cambridge, United Kingdom,*
[3]*Department of Psychology, University of Liverpool, United Kingdom*

Abstract

Colour correction is the process of converting camera dependent RGB values to a camera independent standard colour space such as CIE XYZ. Regression methods — linear, polynomial and root-polynomial least-squares — are traditionally used to solve for the colour correction transform. More recently neural net solutions for colour correction have been developed.

This paper begins with the observation that the neural net solution — while delivering better colour correction accuracy compared to the simple (and widely deployed) 3×3 linear correction matrix approach — is not exposure invariant. That is to say, the network is tuned to mapping RGBs to XYZs for a fixed exposure level and when this exposure level changes, its performance degrades (and it delivers less accurate colour correction compared to the 3x3 matrix approach which is exposure invariant). We develop two remedies to the exposure variation problem. First, we augment the data we use to train the network to include responses for many different exposures. Concomitantly, the trained network is robust to a changing exposure. Second, we redesign the network so, by construction, it is exposure invariant.

Experiments demonstrate that, by adopting either approach, Neural Network colour correction can be made exposure invariant.

Introduction

Colour correction algorithms usually convert camera-related RGB values into camera-independent colour spaces such as sRGB [1] or CIE XYZ [2]. In Figure 1, we plot spectral sensitivity functions of the Nikon D5100 camera and CIE XYZ colour matching functions. If there existed a linear transform which took the Nikon (or any other camera) sensitivity curves so that they were equal the XYZ matching function then the same linear transform would perfectly correct the camera's RGB responses to the corresponding XYZ tristimuli. However, there are no commercial photographic cameras that meet this linear transform condition and so camera RGBs can only be approximately converted to XYZs.

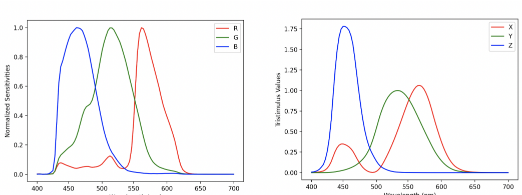

Figure 1. *Normalized sensitivity functions of Nikon D5100 camera (left) and the CIE XYZ standard observer colour matching functions (right). The XYZ matching curves are relative to 'Y' (green curve) which has a maximum response of 1.*

An illustration of the colour correction problem is shown in Figure 2. Here Raw RGB Nikon D5100 camera response are converted by linear colour correction to sRGB [1] colour space. The image shown is drawn from the Foster et al. hyperspectral image set [3] with the RGB and sRGB images calculated by numerical integration. Both images have the sRGB gamma applied.

Figure 2. *The illustration of the colour correction. The images are generated from David Foster's hyperspectral reflectance dataset [3] with Nikon D5100 camera responses and D65 illumination. While the left one demonstrates Raw RGB image, the right one represents the colour corrected sRGB image.*

The most common approach to colour correction maps RGB data to XYZ outputs using s 3×3 matrix (found by regression) such that:

$$M\boldsymbol{\rho} \approx \boldsymbol{x} \tag{1}$$

where $\boldsymbol{\rho}$ and $\boldsymbol{x}$ represents Raw RGB camera response vector and XYZ tristimulus respectively. Polynomial [4] and root-polynomial [5] approaches can also be used for colour correction. In each case an RGB is expanded according to the order of the polynomial (normal or root) and a higher order regression is used to determine the regression transform. As an example the second order root-polynomial expansion maps $[R\ G\ B]^T$ to $[R\ G\ B\ \sqrt{RG}\ \sqrt{RB}\ \sqrt{GB}\ ]^T$ (^{T} denotes transpose) and the correction matrix M is 3×6.

Recently MacDonald and Mayer [6] designed a Neural Net (**NN**) for colour correction and demonstrated that the network delivered colour correction that was better than the linear approach (Equation 1). However, in [7] it was shown that their **NN** approach was not exposure invariant. That is, the network trained to map RGBs to XYZs for a given exposure level delivered relatively poor colour correction when the exposure level changed. The polynomial colour correction algorithm [8] suffers from the same exposure problem: polynomial regression works very well for a fixed exposure but less well when exposure changes [5]. Indeed, this existence of this problem led to the development of the *root* polynomial correction algorithm (which, by construction, is exposure invariant).

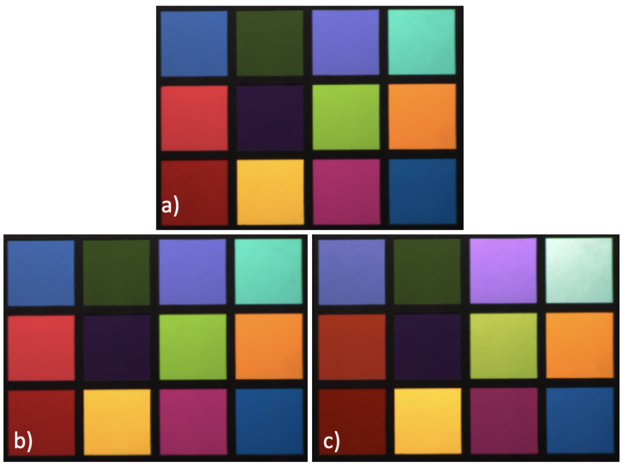

Figure 3. *Top: the true sRGB rendering of 12 coloured patches. Bottom left, the Nikon camera image corrected with a 2nd order polynomial regression. Bottom right shows the output of polynomial colour correction - calculated for exposure = 1 - applied to the camera RGBs * 5 (exposure level = 5). All 3 images are scaled so the brightest value (across all 3 colour channels is 1).*

Let us now run a quick experiment to visually understand the problem of exposure in polynomial regression (where we can get similar results for the Neural Net). For the UEA dataset of spectral reflectance images [9], we randomly sampled 12 reflectances. The actual true sRGB, rendered for a D65 whitepoint, image is shown in the top of Figure 3 (a). On the left of Figure 3 (b) we render these reflectances using the Nikon camera sensitivities and correct the 12 RGBs to the corresponding sRGB values using a 2nd order polynomial expansion. In detail the 2nd order expansion has 9 terms, $[R^2\ G^2\ B^2\ RG\ RB\ GB\ R\ G\ B]^T$, and the colour correction matrix is 9×3. In both the sRGB and fitted camera image, the maximum over all pixel values (across all 3 colour channels) is scaled to be 1.

Now we multiply the Nikon RGBs and the corresponding sRGB triplets by 5. As before, we calculate the 2nd order polynomial expansion of the RGBs and then apply the **same** colour correction matrix for the exposure=1 condition. After colour correction we again scale to make the brightness pixel value (across all 3 channels) equal to 1. The resulting image is shown bottom right of Figure 3 (c). It is clear both that the 'colours' of some patches have changed significantly (see the corresponding reds in the first column and the corresponding cyan in the first row) and that the colour correction is more accurate for the colours rendered under the same exposure conditions (panel (b) is more similar to (a) than (c) is to (a)).

In this paper, we seek to make Neural Network colour correction exposure invariant. We investigate two approaches. First, we augment the training data used to define the neural network with data drawn from many different exposure levels. Second, we design a new network, which, by construction is exposure invariant. Our new network has two components. The chromaticity component network attempts to map camera rgb chromaticity to colorimetric xyz chromaticity. In the second component we linearly correct R, G and B to predict X+Y+Z (mean colorimetric brightness). Given this mean brightness and the target xyz chromaticity we can calculate XYZ. Significantly, we show that the combination of the chromaticity correcting network and the linear brightness predictor generates XYZs in an exposure invariant manner.

Experiments demonstrate that both of our exposure invariant networks continue to deliver better colour correction than a 3×3 linear matrix.

Background

Let $Q_k(\lambda)$ denote the k-th camera spectral response function and $\boldsymbol{Q}(\lambda)$ denote the vector of these functions as in Figure 1. The camera response to a spectral power distribution $E(\lambda)$ illuminating the j-th reflectance $S_j(\lambda)$ is written as:

$$\boldsymbol{\rho} = \int_{\omega} \boldsymbol{Q}(\lambda)E(\lambda)S_j(\lambda)d\lambda \tag{2}$$

where ω denotes the visible spectrum (400 to 700 Nanometres) and $\boldsymbol{\rho}$ denotes the vector of RGB responses. Similarly, given the XYZ colour matching $\boldsymbol{X}(\lambda)$, the tristimulus response $\mathbf{x}$ is written as:

$$\boldsymbol{x} = \int_{\omega} \boldsymbol{X}(\lambda)E(\lambda)S_j(\lambda)d\lambda \tag{3}$$

Suppose, n×3 matrices P and X record (in rows) the camera responses and tristimuli of n surface reflectances, respectively. To find the M in Equation 1 we minimise:

$$\arg\min_{M} \|PM - X\|_F \tag{4}$$

where $\|.\|_F$ denoted the L2 norm [10]. We can solve for M in closed form using the Moore-Penrose Inverse [11]:

$$M = \left[P^T P\right]^{-1} P^T X \tag{5}$$

To extend the regression method we define a basis function $f_e{}^o()$ where the subscript e denotes the type of expansion — here e=p and e=r respectively denotes polynomial and root-polynomial expansions — and the superscript o denotes the order of the expansion. As an example, if we are using the 2nd order root-polynomial expansion [5] then we write:

$$f_r{}^2(\boldsymbol{\rho}) = \left[R\ G\ B\ \sqrt{RG}\ \sqrt{RB}\ \sqrt{GB}\right]^T \tag{6}$$

Again we can use Equations 4 and 5 to solve for the regression matrix M. Though, M will be non-square (and depend on the number of terms in the expansion). For our second order root-polynomial expansion, the columns of P will be the 6 terms in the root-polynomial expansion (P is a n×3 matrix) and M will be 6×3.

In colour correction, we are often interested in how well algorithms work in terms of a perceptually relevant error metric. As examples given the ground-truth sRGB values and their estimated counterparts (delivered by colour correcting camera outputs) we can calculate the colour difference in the CIELAB colour space using the CIE Delta E [2] and CIE Delta E 2000 [12] formulae. However, when a colour difference metric is used, finding the best colour correction transform can no-longer be solved in closed form. Rather, a search based optimisation [7] is used to find regression transform.

As an alternative to regression methods, colour correction can also be implemented as an artificial neural network. MacDonald and Mayer's [6] recently published neural network is illustrated in Figure 4 and is a leading method for neural network colour correction.

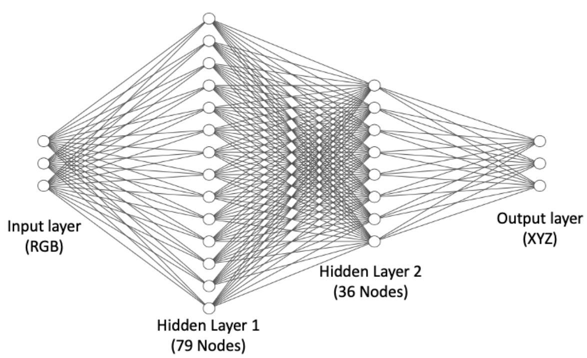

Figure 4. *MacDonald and Mayer's Neural Net [6]. Input and output layers consist of 3 nodes which are RGB and XYZ respectively. In between, there are 2 hidden layers formed by 79 and 36 nodes.*

This Neural Net has 3189 'connections' indicating the cost of colour correction is on the order of 3189 multiplications and additions (the number of operations applied as data flows from left to right). In comparison, the complexity of the 2^{nd} order root-polynomial correction has 3 square root operations and (when the 6×3 correction matrix is applied) 18 multiplications and 15 additions i.e. it is 2 orders of magnitude quicker to compute. In part, the practical utility or otherwise of the Neural Net approach will rest on the trade-off between how well it improves colour correction (say compared to the linear method) against it's higher computational cost.

Exposure Invariant Neural Nets

Abstractly, we can think of a neural network as implementing a vector function $f()$ such that

$$f(\rho) \approx \mathbf{x} \tag{7}$$

When exposure changes — for example if we double the quantity of light — then, physically, the RGB and XYZ responses also double in magnitude. We would like a colour correction function to be exposure invariant:

$$f(k\rho) \approx k\mathbf{x} \tag{8}$$

where k in Equation 8 is a positive scalar. This *homogeneity* property is actually rare in mathematical functions. It holds for linear transforms - $f(\rho) = M\rho$ implies that $f(k\rho) = kM\rho$ - and root-polynomials but it is not true for polynomial expansions [5]. A Neural Net, in order to not collapse to a simple affine transformation, uses non-linear activation functions. These non-linearities, while an essential aspect of the network, make it difficult to attain homogeneity. This homogeneity is exactly what is necessary to achieve good performance over a wide range of exposure levels. As we report in the experimental section, the MacDonald and Mayer network is found not to be exposure invariant.

In Neural Network research if we observe poor performance for some input data then the *trick* is to retrain the network where more of the problematic data is added to the training set. In Neural Network parlance we *augment* the training data set. Here, we have a problem that a network trained for one light level delivers poor colour correction when the light levels changes (e.g. when there is double the light illuminating a scene). So to achieve better colour correction as exposure levels change, we will augment our colour correction training data — corresponding RGBs and XYZs for a single exposure level — with corresponding RGBs and XYZs for several exposure levels. Our retrained, using the exposure level augmented dataset, MacDonald and Mayer Network is our first (more) exposure-invariant neural network solution to colour correction.

Perhaps a more elegant approach to solving the exposure problem is to redesign the network so it is, by construction, exactly exposure invariant. We show such an architecture in Figure 5. In the top network we learn — using Macdonald and Mayer's **NN** — the mapping from input r, g and b chromaticity to x, y and z chromaticity. When the camera and tristimulus response are denoted $[R\ G\ B]^T$ and $[X\ Y\ Z]^T$ then the corresponding chromaticities are defined as $r = R/(R+G+B)$, $g = G/(R+G+B)$ and $b = B/(R+G+B)$; and $x = X/(X+Y+Z)$, $y = Y/(X+Y+Z)$ and $z = Z/(X+Y+Z)$. In the 'intensity' network (bottom of Figure 5) we map R, G and B to predict $X+Y+Z$ by using only a linear activation function. Multiplying the estimated $[x\ y\ z]^T$ by the estimated $X+Y+Z$ returns an estimated $[X\ Y\ Z]^T$.

Informally, let us step through an example to show that the network is exposure invariant. That is we want to show that the respective RGBs ρ and $k\rho$ are mapped to the estimated XYZs $\mathbf{x}$ and $k\mathbf{x}$. Let's consider the RGB vector: [10, 50, 40]. To make the r, g, b chromaticities, we divide RGB values by sum of RGB yielding the r, g, b chromaticities: [0.1, 0.5, 0.4]. Suppose our chromaticity network outputs [0.3, 0.4, 0.5] (the estimates of the x, y, z chromaticities) and the second network (the bottom one in Figure 5) returns 50 as the prediction of X+Y+Z. Now, we multiple output x, y, z chromaticities by 50, we generate the XYZ output: [15, 20, 25].

Now, let's double RGB values: [20, 100, 80]. Clearly, the chromaticities are unchanged ([0.1, 0.5, 0.4]). Because, the output of second network is a simple linear dot-product, output must be equal to 100 (as opposed to 50 before the exposure doubling). Finally, we multiply the estimated x, y, z chromaticities, [0.1, 0.4, 0.5], by 100 and the final output is [30, 40, 50] (which is exactly double as before). This simple example demonstrates that if the

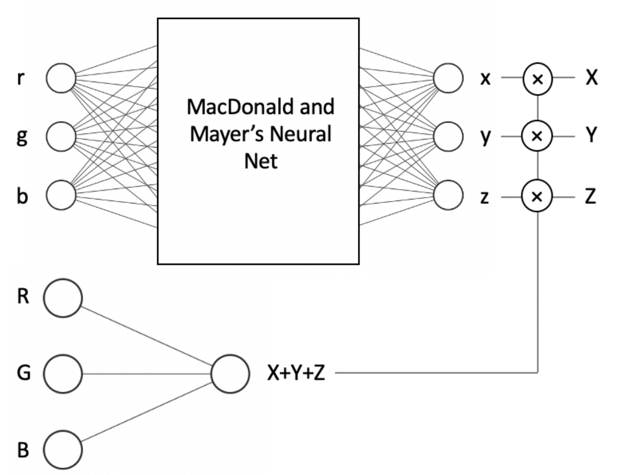

Figure 5. **NN-EI** *architecture with two networks. While the top one learns chromaticities, the bottom one learns sum of XYZs. Multiplications of these gives us the XYZs.*

exposure changes by a scalar k then the output of the network also scales by k and so our new network is exposure invariant.

Experiments

In our experiments, we used the Simon Fraser University (SFU) reflectance set [13] which comprises 1995 spectral surface reflectances including, the 24 Macbeth colour checker patches, 1269 Munsell chips, 120 Dupont paint chips, 170 natural objects, and additional 407 surfaces. The Nikon D5100 camera spectral sensitivities [14] and the viewing illuminant is D65 [15]. All RGBs are calculated using numerical integration. And, the target of colour correction are the corresponding XYZs (also calculated by numerical integration).

The following algorithms are investigated in this paper:

(i) **LS**: Least Squares Regression.

(ii) **LS-RP**: Least Squares Root-Polynomial Regression. Here we use a 2nd order expansion is used.

(iii) **LS-RP-Lab**: Least Squares Root-Polynomial Regression where a search based strategy [16] is used to minimise CIE Delta E error [7].

(iv) **NN**: MacDonald and Mayer's neural net [6].

(v) **NN-AUG**: The **NN** with a augmented training data with different exposure levels.

(vi) **NN-EI**: Here we use two different neural networks. The first one learns to calculate chromaticities and the second one for the sum of XYZ.

As suggested in MacDonald and Meyer's original paper all the colour correction NNs are trained to minimize CIE Delta E 2000 error.

Our colour correction algorithms are tested using a 5 fold cross validation methodology. Here the reflectance dataset is split into 5 equal sized folds (399 reflectances per fold). Each algorithm is trained using four of the folds and then tested on the remaining fold to generate error statistics. The process is repeated 5 times (so every fold is the test set exactly once). According to this methodology, the reported error statistics are averages over the 5 experiments. As an example for each fold (used as a training set), each algorithm will deliver colour correction where a single patch has a maximum correction error. Because we repeat our experiment 5 times the 'maximum error' according to our experiment is the average of the 5 maxima.

Colour correction performance for all algorithms is first calculated for a fixed reference exposure (exposure = 1) level. Then we test our models under different exposure levels to understand their performance when the exposure level changes. We use exposure values of 0.2, 0.5, 1, 2 and 5 (e.g. 0.2 and 5 respectively mean the amount of light is 1/5 and 5 times the reference condition of exposure 1).

In **NN-AUG**, in order to achieve successful results at different exposure levels, we augmented the training data with different exposure factors, which are 0.1, 0.2, 0.5, 2, 5, 10 times the original samples. Then, we tested the models with the test samples with an original exposure level.

Because the **NN-EI** is, by construction, exposure invariant, it is only trained using the data for the reference exposure level.

Details about how the Neural Net was trained

For **NN**, we used MacDonald and Mayer's [6] neural network which has RGBs as input, XYZs as target, the 3x79x36x3 architecture with 2 hidden layers as shown in Figure 4. As in the original paper, we use the Adam optimizer with a learning rate of 0.001 to train the network to minimise CIE Delta E 2000 [12]. We had to raise the number of epochs from 65 (used in the original study) to 500 for the neural network to develop a successful mapping because we were working on relatively small datasets. We also used a mini-batch gradient descent with a batch size of 8. Our model used 20% of the training data for the validation set and used the early stopping method, which means that the training ends automatically if there is no improvement in validation loss after a specified number of epochs (which in our model is 100) with a call-back function. We choose the best model based on the validation loss. The **NN-AUG** and **NN-EI** are trained using the same methodology.

Results

In Tables 1 and 2, we report the CIE Delta E and CIE Delta E 2000 error results, respectively, of 6 algorithms for the fixed exposure level reference condition. The mean, median, max and 95^{th} percentile errors are recorded. Evidently, the neural networks we implemented have better performance than simple least-squares across all error metrics. However, performance is not as good as the Root-Polynomial method — whether found directly as a least-squares computation or by searching — outperforms all the neural networks.

When we compare the colour correction results for the three neural networks, we see that the neural network we trained with different light levels (**NN-AUG**) gives better results than the original neural network (**NN**). We note that the **NN-AUG** neural network is trained with 6 times more data, and this draws attention to the fact that neural networks work better given larger datasets. Regarding this point, the original Macdonald and Mayer network was trained using an even bigger dataset however, neither this data not their trained network are publicly available. It is evident that, for the fixed exposure condition, that the **NN-EI** delivers slightly worse results than the **NN** and **NN-AUG** for a single exposure level, however, it's still better than the standard Least Squares Regression.

Table 1: Mean Delta E statistics

	Mean	Max	Med	95%
LS	1.62	15.47	0.93	5.32
LS-RP	1.19	13.97	0.70	3.62
LS-RP-Lab	1.10	7.36	0.72	3.39
NN	1.40	12.26	0.93	4.06
NN-AUG	1.30	11.00	0.93	3.56
NN-EI	1.53	9.71	1.03	4.40

Table 2: Mean Delta E 2000 statistics

	Mean	Max	Med	95%
LS	0.94	7.71	0.70	2.62
LS-RP	0.72	7.13	0.49	2.14
LS-RP-Lab	0.69	3.81	0.52	1.96
NN	0.85	4.18	0.66	2.15
NN-AUG	0.81	4.37	0.67	1.99
NN-EI	0.95	4.60	0.75	2.29

In Table 3, we report the performance results of the 6 methods at different light levels. As a reminder, in this experiment, the algorithms were trained with the original light level and tested with different intensity levels. The linear and root-polynomial regression methods are unaffected by exposure (they have the same performance across exposure changes). Equally, it is evident that the original MacDonald and Meyer **NN** performs poorly when the exposure changes.

Although the **NN-AUG** method exhibits a fair degree of exposure invariance, its performance still degrades slightly as the change in light levels is more extreme (compared to the reference condition). The **NN-EI** - that was designed to be exactly exposure invariant - delivers better results than the **NN-AUG** method, especially for the conditions where the exposure level is smaller than 0.5 or bigger than 5.

Finally, we see that while the **NN** approach outperforms linear correction the root polynomial method (much simpler with an order of magnitude smaller number of parameters and much less costly training and implementation) performs significantly better.

Table 3: Mean Delta E statistics at different exposure levels

EV	**0.2**	**0.5**	**1**	**2**	**5**
LS	1.62	1.62	1.62	1.62	1.62
LS-RP	1.19	1.19	1.19	1.19	1.19
LS-RP-Lab	1.10	1.10	1.10	1.10	1.10
NN	2.60	1.57	1.40	1.92	3.77
NN-AUG	2.25	1.50	1.30	1.25	1.38
NN-EI	1.53	1.53	1.53	1.53	1.53

Conclusion

Recently, it has been proposed that Neural Networks can be used to solve the colour correction problem. Indeed, in line with previous work found the **NN** approach delivered a modest performance increment compared to the (almost) universally used linear correction method. However, we also found that the **NN** approach was not exposure invariant. Specifically, a network trained for one light could actually deliver poor colour correction as the exposure changed (there was more or less light in the scene).

In this paper, we showed that NNs could be made robust to changes in exposure through data augmentation: by training the NNs with data drawn from many different light levels. In a second approach, we redesigned the neural network architecture so, by construction, it was exactly exposure invariant. Experiments demonstrated that both exposure invariant networks continued to out perform linear colour correction. However, the simple exposure-invariant root-polynomial regression method worked best overall (outperforming the **NN** by about 25%).

Acknowledgments

This research was funded by Spectricity and EPSRC grant EP/S028730.

References

[1] M. Anderson, R. Motta, S. Chandrasekar and M. Stokes, "Proposal for a Standard Default Color Space for The Internet— srgb" Color and Imaging Conference, No. 1, pp. 238-245 (1996).

[2] R. W. G. Hunt and M. R. Pointer, "Measuring colour", John Wiley & Sons, 4th ed. (2011).

[3] D. H. Foster, S. M. C. Nascimento and M. J. Foster, "Frequency of Metamerism in Natural Scenes" Journal of the Optical Society of America A, Vol. 23, pp. 2359-2372 (2006).

[4] G. D. Finlayson and M. S. Drew, "Constrained Least-Squares Regression in Color Spaces" Journal of Electronic Imaging, Vol. 6, No. 4, pp. 484–493 (1997).

[5] G. D. Finlayson, M. Mackiewicz and A. Hurlbert, "Color Correction Using Root-Polynomial Regression" IEEE Transactions on Image Processing, Vol. 24, No. 5 (2015).

[6] L. MacDonald and K. Mayer, "Camera Colour Correction using Neural Networks" London Imaging Meeting, pp. 54-57 (2021).

[7] A. Kucuk, G. D. Finlayson, R. Mantiuk and M. Ashraf, "Comparison of Regression Methods and Neural Networks for Colour Correction" London Imaging Meeting (2022).

[8] G. Hong, M. R. Luo and P. A. Rhodes, "A Study of Digital Camera Colorimetric Characterization Based on Polynomial Modeling" Color Research and Application Vol. 26, No. 1 pp. 76-84 (2001).

[9] S. Hordley, G. D. Finlayson and P. Morovic, "A Multi-Spectral Image Database and Its Application to Image Rendering Across Illumination" Third International Conference on Image and Graphics (ICIG'04). IEEE, pp. 394-397 (2004).

[10] R. A. Horn and C. R. Johnson, "Norms for vectors and matrices" Matrix analysis, pp. 313-386 (1990).

[11] R. Penrose, "A Generalized Inverse for Matrices", Mathematical Proceedings of The Cambridge Philosophical Society, Vol. 51, No. 3, pp. 406-413 (1955).

[12] G. Sharma, W. Wu and E. N. Dalal, "The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations" Color Research and Application pp. 21–30 (2005).

[13] K. Barnard, L. Martin, B. Funt and A. Coath, "A Data Set for Colour Research" Color Research and Application, Vol. 27, No. 3, pp. 147-151 (2002).

[14] M. M. Darrodi, G. D. Finlayson, T. Goodman and M. Mackiewicz, "Reference Data Set for Camera Spectral Sensitivity Estimation" Journal of the Optical Society of America A, Vol. 32, No. 3, pp. 381-391 (2015).

[15] O. Noboru, A. R. Robertson, "3.9: Standard and Supplementary Illuminants". Colorimetry. Wiley. pp. 92–96 (2005).

[16] J. C. Lagarias, J. A. Reeds, M. H. Wright and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions" SIAM Journal of Optimization, Vol. 9, No. 1 pp. 112-147 (1998).