

Exposure Invariance in Spectral Reconstruction from RGB Images

Yi-Tun Lin and Graham D. Finlayson
University of East Anglia, Norwich, United Kingdom

Abstract

In the spectral reconstruction (SR) problem, reflectance and/or radiance spectra are recovered from RGB images. Most of the prior art only attempts to solve this problem for fixed exposure conditions, and this limits the usefulness of these approaches (they can work inside the lab but not in the real world). In this paper, we seek methods that work well even when exposure is unknown or varies across an image, namely ‘exposure invariance’. We begin by re-examining three main approaches - regression, sparse coding and Deep Neural Networks (DNN) - from a varying exposure viewpoint. All three of these approaches are predominantly implemented assuming a fixed capturing condition. However, the leading sparse coding approach (which is almost the best approach overall) is shown to be exposure-invariant, and this teaches that exposure invariance need not come at the cost of poorer overall performance. This result in turn encouraged us to revisit the regression approach. Remarkably, we show that a very simple root-polynomial regression model - which by construction is exposure-invariant - provides competitive performance without any of the complexity inherent in sparse coding or DNNs.

1. Introduction

Spectral information of the scenes is useful in a wide range of applications [15, 4, 13]. Consequently, various imaging technologies have been developed to capture high resolution spectra at each pixel of a scene, namely the *hyperspectral image*. However, these devices are costly and/or bulky and suffer from long integration times, which means they cannot be used for scenes where objects move.

Spectral reconstruction (SR) - also known as *spectral estimation* and *spectral super-resolution* in the literature - is an alternative approach to acquiring hyperspectral information. In SR we attempt to find a mapping which relates fewer-channel sensor responses and high resolution spectra. More precisely, for an RGB camera, the SR problem refers to finding a mapping from each RGB in the image to the corresponding reflectance and/or radiance spectrum.

A simple model of image formation can be written as:

$$x_i = \int_{\Omega} l(\lambda)r(\lambda)s_i(\lambda)d\lambda, \quad (1)$$

where x_i , $l(\lambda)$, $r(\lambda)$ and $s_i(\lambda)$ stand for the i -th camera response, the light’s spectral power distribution, the surface reflectance of the object and the i -th camera sensitivity function. Ω denotes the visible spectrum. With respect to this nomenclature, the spectral reconstruction problem can be written as:

$$SR(\mathbf{x}) \approx \begin{cases} r(\lambda) & \text{(reflectance recovery)} \\ l(\lambda)r(\lambda) & \text{(radiance recovery)} \end{cases} \quad (2)$$

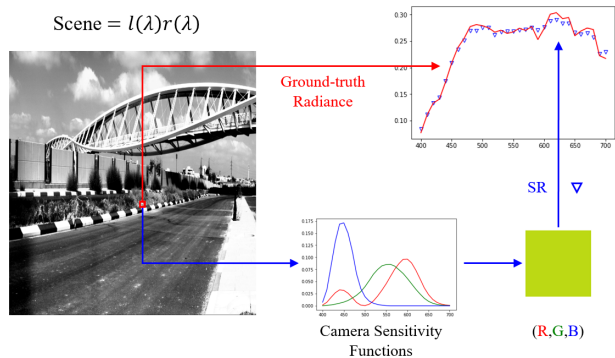


Figure 1. Illustration of spectral reconstruction (SR). The red line in the graph is the true radiance. The mapping from the RGB to the dotted blue line shows the estimated radiance.

where $SR(\cdot)$ denotes a spectral recovery algorithm/function. For one of the images from the hyperspectral data set in [2], we illustrate the SR problem in Figure 1.

Mapping an RGB to the corresponding continuous function of wavelength, either $r(\lambda)$ or $l(\lambda)r(\lambda)$ (respectively the reflectance spectrum and the colour signal or spectral radiance) seems like a hard problem. Yet, it is not as insoluble as one might first suppose. Indeed, the literature from 40 or 50 years ago to-date contains many papers which present methods for solving the problem. The early work concentrated on developing low-dimensional physics-based models for reflectance. Starting with Maloney and Wandell [14], there are many works that characterise reflectances by a 3-dimensional linear model: assuming the illumination spectrum is known, there is a unique 3-dimensional linear transformation mapping RGBs to spectra. Slightly better reflectance recovery was made possible as the problem was posed a least-squares regression to be solved, see [11, 7, 8], or regarded as a Bayesian estimation problem [5, 16].

In the recent literature, the *leading* approaches for SR are sparse coding [2, 1] and (usually deep) Neural Network solutions [17, 12, 20], with the idea of simple regression having been abandoned *due to its simplicity* (the problem is too complex, right?) and that it seems to recover spectra less well. We put *leading* in italics because one of the contributions of this paper is to rehabilitate the regression approach (and cast doubt on the reasons why it was abandoned).

However, whatever approach is used, we notice that the current experimental evaluation framework assumes the capturing condition of the training images is identical to the test images. Yet, the real world may be darker or brighter according to the capturing settings and/or the viewing conditions: the exposure varies between image captures. The exposure also varies *within* an image, e.g. the same physical object can be more or less well exposed in different parts of the scene. In regard to these concerns,

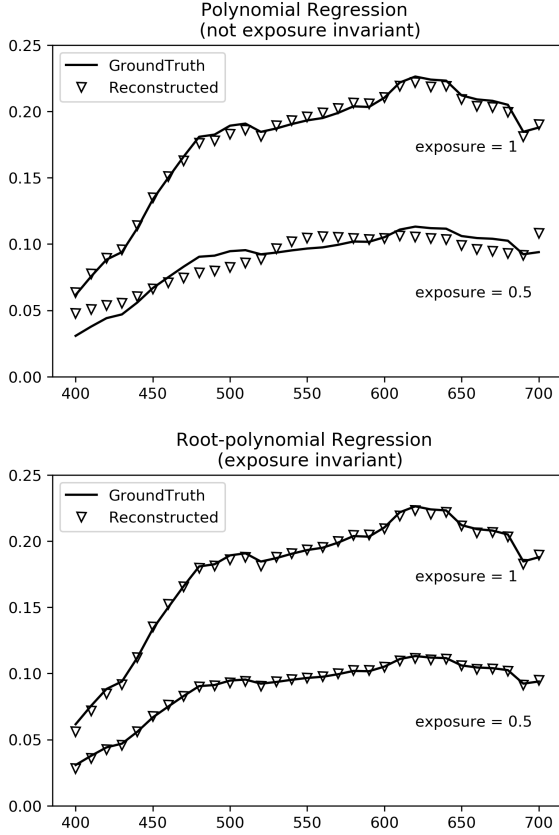


Figure 2. Non exposure-invariant (top) and exposure-invariant SR (bottom)

a key contribution of this paper is to consider spectral reconstruction when the exposure varies.

In Figure 2, we illustrate both the problem and a solution we develop in this paper. The solid lines in the figure show the same ground-truth radiance spectra scaled by a brightness factor, *i.e.* an exposure difference. While the two SR models were calibrated for the longer exposure condition (exposure = 1), the reconstructed spectra (marked as empty triangles) show that non-exposure-invariant polynomial regression (top panel) gets a poorer spectral recovery when the exposure is reduced.

The ‘solution’, which will be later introduced in section 3, is another contribution of this paper: root-polynomial spectral reconstruction. Root-polynomials by construction scales with intensity [9] while general polynomials do not. As shown in the bottom panel of Figure 2, the corresponding recovery for the root-polynomial method is equally good when exposure varies. Moreover, the experimental results in section 4 will demonstrate that root-polynomial regression provides competitive performance compared to even more complex SR methods.

2. Background

2.1. Regression-based Models

Let us denote a single radiance spectrum in the image as a discrete vector \mathbf{r}_j and the corresponding RGB as \mathbf{x}_j , where we assume the former is sampled at k points across the visible spectrum. Here and throughout the paper we assume the visible spectrum runs from 400 to 700 nanometres and we use a 10 nm sampling

distance, so $k = 31$ and \mathbf{r}_j is a 31-vector.

Recall Equation (2), the aim of SR is to map the RGB to the radiance spectrum: $\mathbf{x}_j \mapsto \mathbf{r}_j$. The simplest form of mapping is to conduct a linear transformation by a 31×3 matrix, denoted as the regression matrix \mathbf{M} , such that $\mathbf{M}\mathbf{x}_j \approx \mathbf{r}_j$. To solve for \mathbf{M} , in the least-squares sense, the following expression is to be minimized, given m training pairs of RGBs and spectra:

$$\mathbf{M}_{LR} = \arg \min_{\mathbf{M}} \left(\sum_{j=1}^m \|\mathbf{r}_j - \mathbf{M}\mathbf{x}_j\|_2^2 + \gamma \|\mathbf{M}\|_F^2 \right), \quad (3)$$

where *LR* denotes *linear regression*. Here γ is a user defined parameter as a penalty term. When $\gamma = 0$ the regression is *simple* least-squares. Sometimes the predictions by a regression are *unstable*. That is, regarding the SR problem, small perturbations in RGBs can lead to large changes in the recovered spectra. The aforementioned penalty term γ ‘regularises’ the solution in effect making the regression stable. More details can be found in [11] and [21].

Equation (3) is solved in closed form:

$$\mathbf{M}_{LR} = \mathbf{R}^T \mathbf{X} (\mathbf{X}^T \mathbf{X} + \gamma \mathbf{I}_{3 \times 3})^{-1} \quad (4)$$

where respectively \mathbf{X} and \mathbf{R} are $m \times 3$ and $m \times 31$ matrices (the rows are respectively matched RGBs and spectra) and $\mathbf{I}_{3 \times 3}$ is a 3×3 identity matrix.

A simple way to include non-linearity to the regression model is to apply the *polynomial transformation* to the camera response [7], namely the polynomial regression (*PR*). For an RGB camera, the 2nd, 3rd and 4th degree polynomial feature vectors are defined as follows:

$$\begin{aligned} \Phi^2(\mathbf{x}) &= \left(R, G, B, R^2, G^2, B^2, RG, GB, RB \right)^T \\ \Phi^3(\mathbf{x}) &= \left(R, G, B, R^2, G^2, B^2, RG, GB, RB, \right. \\ &\quad \left. R^3, G^3, B^3, RG^2, GB^2, RB^2, GR^2, BG^2, BR^2, RGB \right)^T \\ \Phi^4(\mathbf{x}) &= \left(R, G, B, R^2, G^2, B^2, RG, GB, RB, \right. \\ &\quad \left. R^3, G^3, B^3, RG^2, GB^2, RB^2, GR^2, BG^2, BR^2, RGB, \right. \\ &\quad \left. R^4, G^4, B^4, R^3G, R^3B, G^3R, G^3B, B^3R, B^3G, \right. \\ &\quad \left. R^2G^2, G^2B^2, R^2B^2, R^2GB, G^2RB, B^2RG \right)^T. \end{aligned} \quad (5)$$

The regression problem of *PR* is defined analogously to Equation (3):

$$\mathbf{M}_{PR}^o = \arg \min_{\mathbf{M}} \left(\sum_{j=1}^m \|\mathbf{r}_j - \mathbf{M}\Phi^o(\mathbf{x}_j)\|_2^2 + \gamma \|\mathbf{M}\|_F^2 \right), \quad (6)$$

where the superscript o denotes the order of the polynomial used. The solution of Equation (6) is written in closed form as:

$$\mathbf{M}_{PR}^o = \mathbf{R}^T \mathbf{X}_{\Phi^o} (\mathbf{X}_{\Phi^o}^T \mathbf{X}_{\Phi^o} + \gamma \mathbf{I}_{n \times n})^{-1} \quad (7)$$

where the i -th row of \mathbf{X}_{Φ^o} is the polynomial expansion of the i -th camera RGB response, and n is the number of terms of the polynomial expansion.

2.2. Sparse Coding

The basic idea behind sparse coding approach is that all spectra can either be found in or as linear combinations of a small dictionary of spectra. Let us suppose the spectral dictionary (D_r) is composed of M radiance spectra:

$$D_r = [\mathbf{r}_1 \ \mathbf{r}_2 \ \dots \ \mathbf{r}_M] . \quad (8)$$

The corresponding RGBs of all items in D_r forms an RGB dictionary:

$$D_x = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_M] . \quad (9)$$

Now let \mathbf{w} denote an M -dimensional weight vector. Since image formation is linear (see Equation (1)), applying the weights to D_r (making a derived spectrum) implies the corresponding RGB of the derived spectrum can be constructed by applying the same weights to D_x :

$$D_r \mathbf{w} = \mathbf{r}' \Rightarrow \mathbf{x}' = D_x \mathbf{w} . \quad (10)$$

It is assumed that the weights in \mathbf{w} are all positive and sum up to 1 and furthermore \mathbf{w} is *sparse*. ‘Sparse’ means that we expect most of the terms in the weight vector to be zero (or very small), such that the fit to a given RGB would only involve a small number of basis vectors.

Sparse coding **SR** operates in two stages. In pre-processing, the spectral dictionary D_r is found (and its optimisation can be laborious). In the second stage we ‘look up’ the dictionary to map RGBs to spectra and this processing is generally faster. The look-up processing in turn operates in three steps. First, the ‘neighbours’ of the concerned RGB response in D_x are found. Second, the combination of the neighbours which matches the RGB is calculated. This combination is then applied to the spectral dictionary D_r to recover the spectral estimate.

2.3 Neural Networks

Neural Networks are well-known for their capability of determining non-linear mappings from input to output vectors. Shallow networks [19, 18, 17] were proposed to find the mapping $SR: \mathbb{R}^3 \mapsto \mathbb{R}^{31}$ which is defined on input of a single camera response vector. More recent approaches consider an image patch as one training/testing sample, for which deep networks such as Convolutional Neural Network (*CNN*) and Generative Adversarial Network (*GAN*) are popular solutions to the problem.

In the recent *NTIRE 2018 Challenge on Spectral Reconstruction from RGB Images* [3], it was reported that all leading models involved implementation of Deep Neural Networks. We select the *CNN*-based HSCNN-D and HSCNN-R models (1st and 2nd place entries in the challenge) [20, 3] for testing in this paper.

A detailed presentation of the various networks is not possible in the restricted space of this paper. But, it is worth remarking that they are often parameterized by millions of parameters and are thus much more complex than regression-based methods.

3. Proposed Method

3.1 Exposure Invariance

Suppose we change the exposure: the amount of light entering the camera (at a pixel). We model this by applying a scaling

factor ξ to Equation (1):

$$\xi x_i = \xi \int_{\Omega} I(\lambda) r(\lambda) s_i(\lambda) d\lambda . \quad (11)$$

This exposure scaling ξ is used to represent anything in the image formation process which affects intensity. Examples include changing the exposure time or the aperture size when an image is captured. Equivalently, the illumination prevailing in the scene could become brighter and darker. Of particular concern to this paper is the idea that the same physical reflectance could appear in different parts of the same scene at different intensities: when two spectra are the same up to a scaling factor, the corresponding RGBs will be related by the same factor and any spectral recovery should also be similarly related.

Abstractly, the **SR** problem - of which we have given examples of regression, sparse coding and Neural Networks - requires us to find a mapping function $SR(\cdot)$ that maps image RGB to the corresponding radiance spectrum:

$$SR(\mathbf{x}_j) \approx \mathbf{r}_j . \quad (12)$$

An **SR** method is exposure-invariant if and only if

$$SR(\xi \mathbf{x}_j) \approx \xi \mathbf{r}_j . \quad (13)$$

3.2 Are Existing Spectral Recovery Methods Invariant to Exposure?

Let us consider the three approaches presented in section 2. It is well known that a straightforward linear transformation is scale-invariant. However, linear regression delivers poor spectral reconstruction. For fixed viewing conditions, polynomial regression can better recover spectra from RGBs (a fact we verify in section 4). But, as discussed in [9] a polynomial regression does not scale with exposure (and as per the example shown in Figure 2 we get different recovery performance for two RGBs which are a scaling apart).

Interestingly, whether or not sparse coding is exposure-invariant depends mostly on how we predict RGBs as a sparse function of the dictionary. Suppose, for example, for a given query RGB \mathbf{x}_q we find the three nearest neighbours in D_x which are the closest in terms of their angular distance to \mathbf{x}_q . Then, if irrespective of the exposure we still find the same three neighbours for $\xi \mathbf{x}_q$, it follows that we will recover the same spectrum scaled by ξ . The leading sparse coding method, called A+ [1], turns out to be exposure-invariant. This result encouraged us to revisit the regression approach and our exposure-invariant regression is presented in the next section.

Neural Network solutions to **SR**, similar to polynomial regression, are also not exposure-invariant. Consider the typical architecture of a single neuron:

$$a' = g(\mathbf{w}^T \mathbf{a} + b) , \quad (14)$$

where \mathbf{a} , \mathbf{w} and b are the inputs, weights and bias, $g(\cdot)$ is called the *activation function* and a' is the output from the neuron. Here $g(\cdot)$ could be a sigmoid function or linear rectification. The offset term alone indicates the neuron will not scale with exposure, *i.e.* if the input is $\xi \mathbf{a}$ the output will not be $\xi a'$. Even when $b = 0$, the use of

common non-linear activation functions (sigmoid, \tanh or radial-basis functions) leads to that the neuron output does not scale with the magnitude of the input. Note that the linear rectification is, in fact, scale-invariant when $b = 0$, but its power of including non-linearity to the network highly depends on the non-zero bias terms. Given this view at the level of a single neuron, we can expect that deep networks - of whatever architectures - can hardly be exposure-invariant by construction.

3.3. Root-Polynomial Spectral Reconstruction

As the early *CNN* model was proposed to deal with the **SR** problem [10], Aeschbacher *et al.* [1] defended the sparse coding models reported in [2, 1], showing that they can reach comparable performance to the *CNN* solution. It was argued in the very paper that sparse coding models have the advantage over deeply learned *CNN* models in terms of time efficiency for training and runtime, clarity of relations between parameters, the amount of training data required and the robustness to overfitting. In the previous part of this paper, we have added another considering factor - exposure invariance, which introduces one more significant advantage of those shallow-learned models (linear regression and A+). Still, speaking of model complexity and efficiency, regression-based approaches are the most fundamental and the simplest. To further improve the performance of linear regression while retaining its exposure-invariant property, we introduce a new polynomial fitting model for **SR**: Root-polynomial Regression (*RPR*).

The proposed method is an extension from the work of Finlayson *et al.* [9] on root-polynomial regression for colour correction. While the colour correction problem considers the mapping from the device-dependent polynomial feature to device-independent tristimulus values CIEXYZ (or display RGB), the **SR** problem seeks to estimate radiance spectra.

In Equation (5) we used the nomenclature Φ to denote the ‘polynomials’, here we denote root-polynomials as $\hat{\Phi}$:

$$\begin{aligned} \hat{\Phi}^2(\mathbf{x}) &= \left(R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{RB} \right)^T \\ \hat{\Phi}^3(\mathbf{x}) &= \left(R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{RB}, \right. \\ &\quad \left. \sqrt[3]{RG^2}, \sqrt[3]{GB^2}, \sqrt[3]{RB^2}, \sqrt[3]{GR^2}, \sqrt[3]{BG^2}, \sqrt[3]{BR^2}, \sqrt[3]{RGB} \right)^T \\ \hat{\Phi}^4(\mathbf{x}) &= \left(R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{RB}, \right. \\ &\quad \left. \sqrt[3]{RG^2}, \sqrt[3]{GB^2}, \sqrt[3]{RB^2}, \sqrt[3]{GR^2}, \sqrt[3]{BG^2}, \sqrt[3]{BR^2}, \sqrt[3]{RGB}, \right. \\ &\quad \left. \sqrt[4]{R^3G}, \sqrt[4]{R^3B}, \sqrt[4]{G^3R}, \sqrt[4]{G^3B}, \sqrt[4]{B^3R}, \sqrt[4]{B^3G}, \right. \\ &\quad \left. \sqrt[4]{R^2GB}, \sqrt[4]{G^2RB}, \sqrt[4]{B^2RG} \right)^T. \end{aligned} \quad (15)$$

It is clear that $\hat{\Phi}^o(\xi\mathbf{x}) = \xi\hat{\Phi}^o(\mathbf{x})$ and so root-polynomial regression is exposure-invariant [9].

Note that compared with *PR*, the number of polynomial terms decreases for *RPR*: the 2nd, 3rd and 4th degree polynomial features for *PR* have 9, 19 and 34 terms, while for *RPR* the numbers are reduced to 6, 13 and 22 terms, respectively.

The closed-form solution of *RPR* can be calculated analogously to Equation (6) and (7):

$$\mathbf{M}_{RPR}^o = \arg \min_{\mathbf{M}} \left(\sum_{j=1}^m \|\mathbf{r}_j - \mathbf{M}\hat{\Phi}^o(\mathbf{x}_j)\|_2^2 + \gamma\|\mathbf{M}\|_F^2 \right), \quad (16)$$

and

$$\mathbf{M}_{RPR}^o = \mathbf{R}^T \mathbf{X}_{\hat{\Phi}^o} (\mathbf{X}_{\hat{\Phi}^o}^T \mathbf{X}_{\hat{\Phi}^o} + \gamma \mathbf{I}_{n \times n})^{-1} \quad (17)$$

where the i -th row of $\mathbf{X}_{\hat{\Phi}^o}$ is the root-polynomial expansion of the i -th camera RGB response.

Finally, we allow a small extension to the *RPR* model where we regress based on a small neighbourhood of pixels. Let us denote the root-polynomial expansion at a pixel (u, v) as $\mathbf{X}_{\hat{\Phi}^o}(u, v)$. Assuming a 3×3 pixel neighbourhood, we have 9 pixel locations $(u-1, v-1), (u-1, v), \dots, (u+1, v), (u+1, v+1)$ with 9 expansions of the form $\mathbf{X}_{\hat{\Phi}^o}(u, v)$. We simply stack these 9 vectors one after another, for the whole training data set, giving a combined neighbourhood representation denoted as $\mathbf{X}_{\hat{\Phi}^o}^{3 \times 3}$, which we use to substitute for $\mathbf{X}_{\hat{\Phi}^o}$ in Equation (17). For instance, per-pixel 2nd order root-polynomial regression matrix \mathbf{M}_{RPR}^2 is 31×6 . When a 3×3 neighbourhood is considered, the regression matrix becomes 31×54 . This additional variation of the model is to move the regression idea (even so gently) toward Deep Neural Networks (whose power is in large part derived from the spatial processing of images).

4. Experiments

Table 1. List of tested models

Model	Approach	Exposure invariant?
HSCNN-D	<i>CNN</i>	no
HSCNN-R	<i>CNN</i>	no
A+ (1x1)	Sparse Coding	yes
A+ (3x3)	Sparse Coding	yes
LR	Regression	yes
PR6	Regression	no
RPR6 (1x1)	Regression	yes
RPR6 (3x3)	Regression	yes

We are going to evaluate spectral recovery for the 8 algorithms listed in Table 1. Note that for both *PR* and *RPR* models, 6th order feature mapping were selected, abbreviated as *PR6* and *RPR6*, respectively. *RPR6* (3×3) denotes the 6th order root-polynomial regression in a 3×3 neighborhood. Likewise, for the A+ algorithm [1] we also include per-pixel (1×1) and per-neighbourhood (3×3) versions.

The hyperspectral image database used for the experiments was ICVL data set [2] (Date of Access: 14 Jan. 2019), which contains 201 hyperspectral images including both indoor and outdoor scenes. The spectral dimension of all spectral data was discretized with 10 *nm* intervals within the visible range [400, 700] *nm* (*i.e.* all spectra are 31-vectors). The corresponding RGB images were simulated by integrating the hyperspectral images with the CIE 1964 color matching functions [6]. All models were tested with original, half and double exposure scalings, which were applied

Table 2. Mean and 95-percentile hyperspectral image reconstruction error in MRAE and MAngE under original, half and double exposure settings (i.e. Exposure = 1, 0.5 and 2). Best results are shown in bold. Note that HSCNN-D and HSCNN-R model are excluded from comparison.

Model	MRAE						MAngE (deg.)					
	Exposure = 1		Exposure = 0.5		Exposure = 2		Exposure = 1		Exposure = 0.5		Exposure = 2	
	Mean	95pt.	Mean	95pt.	Mean	95pt.	Mean	95pt.	Mean	95pt.	Mean	95pt.
HSCNN-D	0.0118	0.0161	0.1470	0.2359	0.0577	0.0965	0.78	1.27	7.03	9.25	3.56	6.38
HSCNN-R	0.0130	0.0189	0.1791	0.3655	0.0575	0.0967	0.86	1.42	6.43	8.23	3.52	6.35
A+ (1x1)	0.0385	0.0800	0.0385	0.0800	0.0385	0.0800	2.38	5.16	2.38	5.16	2.38	5.16
A+ (3x3)	0.0432	0.0839	0.0432	0.0839	0.0432	0.0839	2.67	5.31	2.67	5.31	2.67	5.31
LR	0.0624	0.1214	0.0624	0.1214	0.0624	0.1214	3.79	7.47	3.79	7.47	3.79	7.47
PR6	0.0195	0.0296	0.0960	0.1197	0.1300	0.3695	1.46	2.50	5.50	6.69	4.89	9.07
RPR6 (1x1)	0.0470	0.0983	0.0470	0.0983	0.0470	0.0983	2.88	5.68	2.88	5.68	2.88	5.68
RPR6 (3x3)	0.0471	0.0973	0.0471	0.0973	0.0471	0.0973	2.87	5.57	2.87	5.57	2.87	5.57

to the ground-truth hyperspectral data during the testing stage but *not* in the training stage (same trained models were tested for different exposure settings). The Tikhonov regularisation factors in all regression models are found by grid search in logarithmic scale on validation data.

The experiments were conducted with 2-fold cross validation; except for the *CNN* models, the pre-trained networks provided in [20] were directly used. Since the whole ICVL data set was part of their training set (as reported in [3]), the results of HSCNN-D and HSCNN-R shown in this paper are only for demonstrating how they perform in presence of exposure scaling and should not be compared to the other results.

4.1. Exposure-Invariant Metrics

As pointed out in [3], various metrics have appeared in the literature for evaluating the performance of **SR** models. Some of which even have multiple definitions (e.g. RMSE in [1] and [10]). In this paper we aim to compare the model performance under different scene exposures; consequently, the error metrics should not depend on the overall intensity of the pair of spectra that are being compared. The metrics defined below are used.

- Mean Relative Absolute Error

$$MRAE = \frac{1}{N} \sum_i \left\| \frac{\mathbf{r}_i - \hat{\mathbf{r}}_i}{\mathbf{r}_i} \right\|_1 \quad (18)$$

- Mean Angular Error (unit: *degree*)

$$MAngE = \frac{1}{N} \sum_i \text{angle}(\mathbf{r}_i, \hat{\mathbf{r}}_i) \quad (19)$$

Above, \mathbf{r}_i and $\hat{\mathbf{r}}_i$ denote the i -th ground-truth and estimated spectra. In (18), the division in *MRAE* is component-wise and the absolute deviation (L_1 norm) is calculated. In (19) the mean spectral angular error is calculated. Both error measures are calculated per image of N spectra.

4.2. Results and Discussion

The performance statistics in Table 2 show several results. First, for fixed exposure the *CNN* approach works well. The root-polynomial approach introduced in this paper works better than linear regression but not as good as the best sparse coding model

and straightforward polynomial regression: the polynomial regression performs the best overall. If we allow exposure to vary, then the exposure-fixed *CNN* models work poorly. Indeed, they seem highly tuned to fixed exposure condition. Remarkably, those exposure invariant models, including linear, root-polynomial regression and sparse coding, works much better than polynomial regression and most possibly the *CNN* solutions - the leading models under fixed exposure. Lastly, it appears that incorporating small neighbourhood information into root-polynomial regression and A+ does not help.

In Figure 3, we show the error maps in terms of Relative Absolute Error (i.e. *MRAE* without the pixel-wise averaging) for a single image under 3 exposures recovered by the 8 selected **SR** models. The exposure invariance of root-polynomial, linear regression and A+ is clear, while the fact that the *CNN* approaches are highly tuned to the exposure condition is also evident.

We admit the possibility that Deep Neural Networks might be retrained in an exposure-invariant way, e.g. by adding exposure variation to the training regime. We also remark that the recovery error for the root-polynomial method is still good when exposure conditions are held fixed. Indeed, the angular error is less than 3 degrees on average. It could be that a 3 degree error is sufficient for tasks such as material recognition. Testing both of these hypotheses is for future research.

5. Conclusion

Spectral reconstruction (**SR**) algorithms attempt to map RGBs to corresponding spectra. Many of the early **SR** algorithms were based on regression, but these have *apparently* been superseded by more advanced statistical approaches including sparse coding techniques and most recently by Deep Neural Networks (*DNN*). *DNNs* seem to provide a step change in performance.

In this paper we threw exposure variation into the mix. We asked how tuned existing techniques were to the exposure conditions: if we scale RGBs by 0.5 (a physically accurate simulation of dimming the light intensity by 50%) would existing methods still work? We make two important contributions. First we show that the performance of many legacy methods are highly dependent on exposure. For example, two of the best *DNN* approaches work poorly when exposure changes. Second, we propose the root-polynomial regression [9] technique - used in the colour correction literature - to the problem of spectral re-

