

Article

On the Optimization of Regression-Based Spectral Reconstruction [†]

Yi-Tun Lin ^{*} and Graham D. Finlayson

School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK; G.Finlayson@uea.ac.uk

^{*} Correspondence: Yi-Tun.Lin@uea.ac.uk[†] This paper is an extended version of our papers published in (1) Lin Y.T. and Finlayson G.D. Per-channel regularization for regression-based spectral reconstruction. In Proceedings of the 10th Colour and Visual Computing Symposium, CEUR Workshop Proceedings, Gjøvik, Norway, 16–17 September 2020 (online open access: <http://ceur-ws.org/Vol-2688/>). (2) Lin Y.T. and Finlayson G.D. Reconstructing spectra from RGB images by relative error least squares regression. In Proceedings of the 28th Color and Imaging Conference, Online, 4–19 November 2020; pp. 264–269.

Abstract: Spectral reconstruction (SR) algorithms attempt to recover hyperspectral information from RGB camera responses. Recently, the most common metric for evaluating the performance of SR algorithms is the Mean Relative Absolute Error (MRAE)—an ℓ_1 relative error (also known as percentage error). Unsurprisingly, the leading algorithms based on Deep Neural Networks (DNN) are trained and tested using the MRAE metric. In contrast, the much simpler regression-based methods (which actually can work tolerably well) are trained to optimize a generic Root Mean Square Error (RMSE) and then tested in MRAE. Another issue with the regression methods is—because in SR the linear systems are large and ill-posed—that they are necessarily solved using regularization. However, hitherto the regularization has been applied at a spectrum level, whereas in MRAE the errors are measured per wavelength (i.e., per spectral channel) and then averaged. The two aims of this paper are, first, to reformulate the simple regressions so that they minimize a relative error metric in training—we formulate both ℓ_2 and ℓ_1 relative error variants where the latter is MRAE—and, second, we adopt a per-channel regularization strategy. Together, our modifications to how the regressions are formulated and solved leads to up to a 14% increment in mean performance and up to 17% in worst-case performance (measured with MRAE). Importantly, our best result narrows the gap between the regression approaches and the leading DNN model to around 8% in mean accuracy.

Keywords: spectral reconstruction; hyperspectral imaging; multispectral imaging; regression; regularization; inverse problem



Citation: Lin, Y.-T.; Finlayson, G.D. On the Optimization of Regression-Based Spectral Reconstruction. *Sensors* **2021**, *21*, 5586. <https://doi.org/10.3390/s21165586>

Academic Editors: Pierre-Jean Lapray, Jean-Baptiste Thomas and Yusuke Monno

Received: 16 June 2021

Accepted: 15 August 2021

Published: 19 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A consumer RGB camera captures light signals with three types of color sensors. Yet the light is physical radiance—a continuous spectral function of wavelength—which, intuitively, can hardly be described by a 3-dimensional color representation. Indeed, many researchers have found that real-world spectra should be at least 5- to 8-dimensional [1–5]. Consequently, with RGB imaging we can only acquire limited information encoded in the light spectrum.

Using a hyperspectral camera [6–19], we can record scene radiance at high spectral and spatial resolution. This technique has been widely used in machine vision applications such as remote sensing [20–27], medical imaging [28–31], food processing [32–37], and anomaly detection [38–44], as well as in the spectral characterization domain, including the calibration of color devices (e.g., cameras [45] and scanners [46]), scene relighting [47,48], and art conservation and archiving [49–51]. While useful, hyperspectral cameras are usually much more expensive than the RGB cameras. Moreover, the extra spectral

information is often captured with reduced spatial and/or temporal resolution, which limits their usefulness.

Spectral reconstruction (SR) seeks to recover spectral information from the RGB data of a single camera [52–66]. Assuming the recovery error of SR is low enough (regarding the results in the literature and in this paper), we can essentially measure spectra using an RGB camera.

Historically, SR is most efficiently solved by Linear Regression (LR) [52], where the map from RGBs to spectra is described by a simple linear transformation. Considering a nonlinear map, the Polynomial Regression (PR) [53] and Root-Polynomial Regression (RPR) [56] methods expand the RGBs into a set of polynomial/root-polynomial terms—which are then mapped to spectra via a linear transform. More recent regression models, including the Radial Basis Function Network (RBFN) [54] and the A+ sparse coding algorithm [55], use clustering techniques to define the local neighborhood (in the color or spectral space) in which each RGB is regressed.

On the other hand, we see most of the recent SR algorithms are not based on simple regressions but based on Deep Neural Networks (DNN) [60–66] which embrace the idea of regressing RGB image patches as a whole (as oppose to regressing each pixel independently in regressions). This approach hypothesizes that object-level descriptions of the RGBs can—though requiring much more computational resources—aid the recovery of spectra. However, DNN-based models do not always perform better than simple regressions [55] and often suffer from instability issue when recovering spectra at different brightness scales [56–58]. Furthermore, spectra recovered by DNNs are shown to be less accurate in color [57,59] and do not necessarily provide more accurate cross-illumination or cross-device color reproductions [57].

In this paper, we aim to further optimize the existing regression-based SR methods. First, we noticed that in recent works, DNN-based models are evaluated and ranked by Mean Relative Absolute Error (MRAE) [63,64]. Most top DNN models are also designed to minimize MRAE directly [60–64]. However, all regressions used in SR are optimized by convention for Root Mean Square Error (RMSE) while evaluated using MRAE (or other similar relative errors) when compared with the DNNs [55–57,60,61]. In other words, these regressions are optimized for one metric, and then how well they work is evaluated with another.

In Figure 1, we illustrate the standard experimental framework of SR. In training, the parameters of the SR model are tuned such that the “losses”—the differences between the ground-truths and estimations measured by a given loss metric—are statistically minimized. After the SR models are trained, we evaluate them based on a desired evaluation metric. Ideally, the loss and evaluation metrics should match (i.e., the same or similar in nature). Indeed, a model that is optimized for one metric but evaluated by another will surely lead to sub-optimal results.

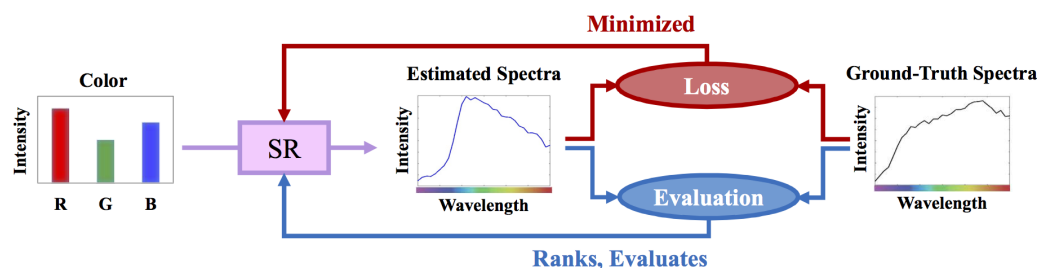


Figure 1. The standard spectral reconstruction (SR) training and evaluation scheme.

Based on this insight, we propose two new minimization approaches for simple regressions—the Relative Error Least Squares (RELS) and Relative Error Least Absolute Deviation (RELAD). While the former minimizes an error similar to MRAE and is solved in closed form, the latter explicitly minimizes the MRAE metric but has the disadvantage of requiring an iterative minimization.

The second contribution of this paper is to propose a new way of regularizing the regression-based spectral reconstruction. Most regressions are necessarily trained using a regularization constraint [67], both to prevent overfitting and to make the system equations more “stable” [68] (a system of equations is stable if small perturbations appear in the training data results in a small perturbation in the solved-for model parameters). However, we observe that hitherto in regression-based spectral reconstruction all spectral channels are regularized altogether—e.g., in [52–56]. That is the regularization constraint is effectively applied at the spectrum level. Yet, fundamentally, error metrics such as MRAE measure the errors at all spectral channels independently and then average them to give a spectral error measure. Thus, we propose a “per-channel” regularization methodology which ensures optimized regularization for each spectral channel independently.

Combined, we find that training the simple regressions to minimize the same error as used in testing and adopting a per-channel regularization approach lead to a significant uptick in performance. Moreover, as shown in the example hyperspectral image reconstruction results in Figure 2, our new regression methods deliver performance that is similar to one of the best (but much more complex) DNN approaches.

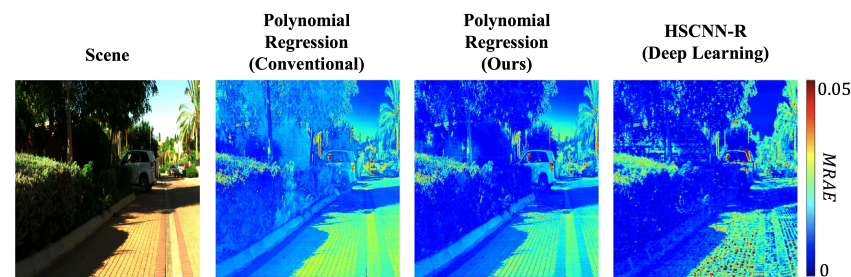


Figure 2. Example hyperspectral image reconstruction error heat maps (in MRAE) by the conventional Polynomial Regression [53] (left), Polynomial Regression based on our new RELS method (middle), and the deep learning-based HSCNN-R [60] (right).

The rest of the paper is organized as follows. In Section 2, we will introduce the prerequisites of regression-based spectral reconstruction. Our new methods are introduced in Section 3. Section 4 details the experimental procedures, and Section 5 discusses the experimental results. This paper concludes in Section 6.

2. Background

2.1. Regression-Based Spectral Reconstruction

A simple mathematical model of RGB formation is written as [69]

$$\underline{x} = \int_{\Omega} r(\lambda) \underline{s}(\lambda) d\lambda, \quad (1)$$

where $r(\lambda)$ denotes the radiance spectrum, $\underline{s}(\lambda) = [s_R(\lambda), s_G(\lambda), s_B(\lambda)]^T$ is the set of three spectral sensitivities of the color sensors, and $\underline{x} = [R \ G \ B]^T$ is the derived color vector (the superscript T denotes the transpose operator). In the case of color imaging, the range of integration, Ω , is the visible range. In this paper, we consider this range to be the interval from 400 to 700 nanometers (nm).

Further, we assume $r(\lambda)$ and $\underline{s}(\lambda)$ are measured by a hyperspectral device at every 10 nm (in accordance with the recent SR challenge datasets [63,64]). As such, we can approximate the integration in Equation (1) by inner products [69]:

$$\underline{x} = \mathbf{S}^T \underline{r}, \quad (2)$$

where $\underline{r} = [r(400), r(410), \dots, r(700)]^T$ is the 31-component vectorized spectrum, and $\mathbf{S}$ is the 31×3 spectral sensitivity matrix (one spectral sensitivity per column).

In SR, we study the inverse problem: we wish to recover $\mathbf{r}$ using the information of $\mathbf{x}$. A general definition of SR (considering both regressions and DNNs) can be written as

$$\Psi(\mathcal{N}_k) \approx \mathbf{r}_k, \mathcal{N}_k = \{\mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \dots, \mathbf{x}_k, \dots, \mathbf{x}_{k,n}\}, \quad (3)$$

where $\mathbf{r}_k$ denotes the hyperspectral data at the location of interest (the k th pixel), Ψ denotes an SR algorithm, and the neighborhood set $\mathcal{N}_k$ includes the RGBs in the spatial neighborhood of pixel k (the subscripts k,l indexing the l th pixel in the proximity of pixel k). Note that $\mathcal{N}_k$ also contains the RGB at the location of interest i.e., $\mathbf{x}_k$.

In regression we normally do not use the neighborhood information to recover spectra (though small neighborhoods are occasionally admitted [55,56]). Rather, we attempt to map each RGB uniquely to a corresponding spectrum, i.e., $\Psi(\mathbf{x}_k) \approx \mathbf{r}_k$.

2.1.1. Linear Regression

The simplest Linear Regression (LR) approach [52] models the SR algorithm Ψ by a linear transformation:

$$\Psi(\mathbf{x}_k) = \mathbf{M}^T \mathbf{x}_k \approx \mathbf{r}_k, \quad (4)$$

where $\mathbf{M}^T$ is a 31×3 matrix.

Considering the whole database of N spectra and their matching RGBs (regardless of which images they are in and their pixel locations), let us arrange them into the $N \times 31$ spectral data matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k, \dots, \mathbf{r}_N]^T$ and the matching RGB data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k, \dots, \mathbf{x}_N]^T$. Based on this nomenclature, we can rewrite Equation (4) as

$$\mathbf{X}\mathbf{M} \approx \mathbf{R}. \quad (5)$$

That is, all spectra (the rows of $\mathbf{R}$) are estimated from their corresponding RGBs (the rows of $\mathbf{X}$) using the same linear transformation matrix $\mathbf{M}$.

2.1.2. Nonlinear Regressions

The regression problem summarized in Equation (5) is, by construction, linear. In a general case where we may wish to pose the SR problem as a non-linear regression, we might transform each row of $\mathbf{X}$ (each RGB) to an s -component counterpart using a given nonlinear function $\varphi: \mathbb{R}^3 \mapsto \mathbb{R}^s$:

$$\mathbf{X}_\varphi = [\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \dots, \varphi(\mathbf{x}_k), \dots, \varphi(\mathbf{x}_N)]^T. \quad (6)$$

This $N \times s$ transformed RGB matrix is then regressed to estimate the spectral data $\mathbf{R}$ using a new linear transformation matrix $\mathbf{M}_\varphi$:

$$\mathbf{X}_\varphi \mathbf{M}_\varphi \approx \mathbf{R}. \quad (7)$$

Here, the dimension of $\mathbf{M}_\varphi$ is $s \times 31$.

Examples of nonlinear regressions in the literature include Polynomial Regression (PR) [53], Root-Polynomial Regression (RPR) [56], and Radial Basis Function Network (RBFN) [54]. For instance, if we consider the 2nd-order RPR, we will have $\varphi(\mathbf{x}) = [R \ G \ B \ \sqrt{RG} \ \sqrt{GB} \ \sqrt{BR}]^T$, in which case $\mathbf{X}_\varphi$ is an $N \times 6$ matrix and the regression matrix $\mathbf{M}_\varphi$ is 6×31 [56].

2.1.3. A+ Sparse Coding

So far, both linear and nonlinear regressions assume a single regression matrix that is used to estimate spectra for all RGBs. Yet in A+ sparse coding (i.e., adjusted anchored neighborhood regression) [55], it is considered that the SR mapping should be done locally [55,59]. Fundamentally, in sparse coding one assumes that all spectra can be represented by linear combinations of a small number of anchor spectra (hence it is “sparse”) [55,70].

In training, the A+ algorithm runs the K-SVD clustering algorithm [71] and assigns the clusters' centers as the set of anchor spectra. Around the corresponding RGB value of each anchor, a fixed number of nearest RGB neighbors in the training dataset and their corresponding radiance spectra are recorded. These neighboring data points are then used to create a bespoke Linear Regression mapping (Equation (5)) that should be only used by the RGBs in the testing set that are around the same neighborhood [59]. That is, in the reconstruction stage of A+, we first find the anchor spectrum whose matching RGB is closest to the query RGB, and second we map the query RGB to spectrum using the regression mapping associated with that neighborhood.

We note that all the discussed regression models can be formulated by either Equations (5) or (7), while the structure of both are essentially the same. Thus, to simplify the notation we will adopt the regression notation of Equation (5), i.e., we drop the subscript φ of Equation (7). As such, in later discussions the data we are regressing can be the RGBs or their non-linear expansions thereof.

2.2. Least-Squares Minimization

Now let us consider how we solve for the regression matrix $\mathbf{M}$. In essence, we must define what we meant by the $\approx$ symbol in Equation (5).

Most commonly, we seek to minimize the “sum-of-squares” between the estimations and ground-truths:

$$\min_{\mathbf{M}} \|\mathbf{X}\mathbf{M} - \mathbf{R}\|_2^2. \quad (8)$$

Here, $\|\cdot\|_2^2$ calculates the sum of all components squared. Equivalently, $\|\cdot\|_2^2$ is sometimes written as $\|\cdot\|_F^2$ denoting the Frobenius norm.

Regressions solved in this manner are called the Least-Squares (LS) regression. Advantageously, the solution of LS minimization can be solved in closed form [72].

2.2.1. The Overfitting Problem and Regularized Least Squares

However, the solution of Equation (8) often cannot be directly used in practice—given that the solved regression matrix $\mathbf{M}$ is optimized for minimizing the errors in the training set, we often find that this matrix does not work for unseen data. In other words, denoting a set of unseen RGB and spectral data as $\mathbf{X}'$ and $\mathbf{R}'$, we often get $\mathbf{X}'\mathbf{M} \not\approx \mathbf{R}'$. This problem is so-called “overfitting” [67,68].

One of the biggest problems for an overfitted regression is that it might not work in the presence of noise. As a thought experiment, we may consider to recover $\mathbf{R}$ (the same set of training spectral data) from the “perturbed” training RGB data, $\mathbf{X}' = \mathbf{X} + \epsilon$, where ϵ is a matrix of very small numbers, representing the noise occurs in the RGB imaging process. It follows that an overfitted $\mathbf{M}$ can very possibly suggest $\mathbf{X}'\mathbf{M} = [\mathbf{X} + \epsilon]\mathbf{M} \not\approx \mathbf{R}$, that is, it fails to plausibly recover $\mathbf{R}$.

Another facet of this problem is that if we actually attempt to find the best regression matrix for the noisy data $[\mathbf{X} + \epsilon]$, we can end up having the optimal regression matrix that is very different from the original one. That is, as we perturb our RGB data, we can arrive at very different regressions.

To mitigate the overfitting problem, the tool of ridge regularization (a.k.a. Tikhonov regularization) [67] is often incorporated when training a regression. While solving the minimization in Equation (8), we add another penalty term which bounds the magnitude of the regression matrix $\mathbf{M}$:

$$\min_{\mathbf{M}} \|\mathbf{X}\mathbf{M} - \mathbf{R}\|_2^2 + \gamma \|\mathbf{M}\|_2^2, \quad (9)$$

where the solution of $\mathbf{M}$ can still be written in closed form [52].

Here, the γ parameter—which is set by the user—controls how $\|\mathbf{M}\|_2^2$ mitigates the minimization of the sum-of-squares fitting error. Clearly, if $\gamma = 0$, the equation resorts to the simple LS (Equation (8)), and as γ becomes large, the need to solve for an $\mathbf{M}$ that has a small

(bounded) norm becomes imperative and the requirement to reduce the sum-of-squares fitting error is less important.

When the solved $\mathbf{M}$ has a bounded norm, the regression has the stability property we desire. That is, if we perturb $\mathbf{X}$ by a small amount we will still get the same (or very similar) $\mathbf{M}$, and this in turn implies that albeit the perturbation in the input RGBs we will still get similar spectral estimations. We refer the readers to the work in [67] for a fuller discussion of how regularization is used and why it solves the instability problem.

In practice, we choose the γ parameter empirically to best trade-off the need for a stable solution and to lower the fitting spectral errors. Figure 3 illustrates a typical cross-validation parameter selection methodology [73]. First, a wide range of different γ 's are tried, and the solved $\mathbf{M}$'s depending on these γ values are used to recover spectra in an unseen “validation set”, which are then evaluated using the desired evaluation metric (here, we use the MRAE metric). Usually, a “U-shaped” curve is obtained when plotting the averaged validation-set MRAE against γ , where the minimal point (the red dot in the plot) indicates the selected parameter.

We note that the regularization parameter (the red dot in Figure 3) is not “fixed”, that is we search for independent regularization parameter in every minimization instance encountered.

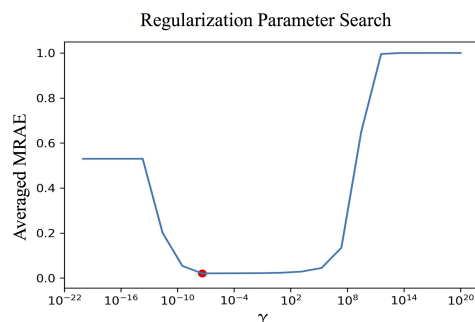


Figure 3. An illustration of searching for the regularization parameter. The averaged MRAE (vertical axis) is calculated over the validation set (a separate data set that is not the training set). The red dot in the graph indicates the minimal error and the suggested regularization parameter.

While the ridge regularization approach introduced here (adding an ℓ_2 penalty term) is the most widely used approach (not only for SR but for learning problems in general), there are other regularization approaches used in the literature, including the LASSO [74] (where the ℓ_1 penalty term is used) and Elastic Net approach [75] (where both ℓ_1 and ℓ_2 penalty terms are added, with dependent or independent regularization parameters). It is known that LASSO provides more robust regularization compared to the ℓ_2 ridge regularization, while the Elastic Net provides the possibility to swing more towards LASSO or ridge regression depending on the relative dominance between the two methods in each specific learning circumstances. For more information of these two methods, interested readers are pointed to the works in [74,75], respectively.

2.3. RMSE versus MRAE Error Measures

In the prior art of SR, most (if not all) regression-based methods are based on LS minimization with a ridge regularization setting (see, e.g., in [52–56]). As the LS minimization uses sum-of-squares as its loss metric (see Equations (8) and (9)), a natural metric for evaluating LS-based regressions is Root Mean Square Error (RMSE):

$$\text{RMSE}(\hat{\mathbf{r}}, \mathbf{r}) = \sqrt{\frac{1}{31} \|\hat{\mathbf{r}} - \mathbf{r}\|_2^2} \quad (10)$$

(because the sum-of-squares is essentially the sum of squared RMSE). Here, we denote $\hat{\mathbf{r}} = \Psi(\mathbf{x})$ as the estimation (returned by, e.g., regressions or DNNs) of the ground-truth $\mathbf{r}$.

Of course, the physical radiance spectra may have greater or lesser magnitudes depending on the imaging conditions [56]. However, if we model the change in brightness by a scalar k , we get $\text{RMSE}(k\hat{\mathbf{r}}, k\mathbf{r}) = k\text{RMSE}(\hat{\mathbf{r}}, \mathbf{r})$. Which means, for a ground-truth spectrum that is k times brighter than $\mathbf{r}$, when compared with the recovered spectrum also scaled by k (i.e., $k\hat{\mathbf{r}}$), we will get k times larger RMSE than the original case.

This biased nature of RMSE evaluation can as well influence the training process of regressions. Indeed, we can expect that the standard LS minimization can overestimate the squared-RMSE (i.e., sum-of-squares) losses of the bright spectra and consequently place more importance on minimizing their errors compared to the dim ones.

This brightness bias is part of the motivation underpinning the Mean Relative Absolute Error (MRAE) metric for spectral recovery [63,64]. This MRAE metric almost universally adopted in the recent literature is defined as

$$\text{MRAE}(\hat{\mathbf{r}}, \mathbf{r}) = \frac{1}{31} \left\| \frac{\hat{\mathbf{r}} - \mathbf{r}}{\mathbf{r}} \right\|_1, \quad (11)$$

where the spectral difference is divided by (component-wise) the ground-truth spectral values. As such, the effect of brightness level is discounted when measuring spectral differences, i.e., $\text{MRAE}(k\hat{\mathbf{r}}, k\mathbf{r}) = \text{MRAE}(\hat{\mathbf{r}}, \mathbf{r})$.

Arguably, the MRAE makes more sense as a performance metric as the SR algorithm might be used where the absolute/maximum intensity of the tested spectra is not known or controlled. Therefore, the main goal of this paper is to consider how to modify the SR regression to minimize a relative recovery error (in general and MRAE in particular).

We would also like to emphasize that the development of our new methods (presented in the following sections) are, of course, tailored to the assumption that MRAE is used as the evaluation error metric. Moreover, again, MRAE is certainly the most commonly used metric adopted for DNNs in the literature at this time, see, e.g., in [60,61,63,64].

3. Rethinking the Optimization of Regression for Spectral Reconstruction

The first part of our contributions concerns the regularization method used in the standard case (Equation (9)), where the regularization takes place at the spectrum level with all spectral channels being regularized together. Here, we will argue—and develop the requisite mathematics—that the regularization should be done per spectral channel.

In our second contribution, we alter the form of the regression. Following that recently MRAE is used to evaluate and rank new methods, and that this metric is optimized directly in modern DNN solutions but not in regression-based methods, we reformulate the regressions to optimize for fitting errors that are “relative” in nature (spectral differences relative to the ground-truth values). We develop solutions based on both ℓ_2 and ℓ_1 relative error minimizations.

3.1. Per-Channel Regularization

Returning to the conventional regression formulation in Equation (5), let us split the regression matrix $\mathbf{M}$ and the ground-truth spectral data $\mathbf{R}$ by columns:

$$\mathbf{X}[\underline{\mathbf{m}}_1, \underline{\mathbf{m}}_2, \dots, \underline{\mathbf{m}}_i, \dots, \underline{\mathbf{m}}_{31}] \approx [\underline{\boldsymbol{\rho}}_1, \underline{\boldsymbol{\rho}}_2, \dots, \underline{\boldsymbol{\rho}}_i, \dots, \underline{\boldsymbol{\rho}}_{31}]. \quad (12)$$

Here, $\underline{\boldsymbol{\rho}}_i$ denotes the i th column of $\mathbf{R}$. Remember that each row of $\mathbf{R}$ is a single radiance spectrum, thus the numbers in $\underline{\boldsymbol{\rho}}_i$ are the i th-channel spectral intensities of all the spectra in the database.

Now, we observe that the original multivariate regression is in fact a collection of 31 independent regressions:

$$\mathbf{X}\underline{\mathbf{m}}_i \approx \underline{\boldsymbol{\rho}}_i \quad \text{for } i = 1, 2, \dots, 31. \quad (13)$$

Notice that Equation (13) is equivalent to the original formulation in Equation (5), only that it explicitly shows there is no “inter-channel” dependence. In other words, for all regression-based SR in the literature, it has been always the case that each column of $\mathbf{M}$ is only used for recovering the corresponding column of $\mathbf{R}$ while irrelevant to the recoveries for other spectral channels.

Curiously, as we solve for $\mathbf{M}$ using the standard LS minimization in Equation (9), the strength of the penalty term, $\gamma \|\mathbf{M}\|_2^2$, is controlled by a single parameter γ , meaning that all columns of $\mathbf{M}$, that is, the 31 $\underline{m}_i$'s, are regularized using the same γ parameter despite the fact that each of them works independently of others. Essentially, by regularizing $\mathbf{M}$ as a whole, we are “asserting” such an interdependence among channels.

From a mathematical viewpoint (regarding how the regression is formulated), it makes more sense to regularize each per-channel regression independently. Following Equation (13), we rewrite Equation (9) in a per-channel fashion:

$$\min_{\underline{m}_i} \|\mathbf{X}\underline{m}_i - \underline{\rho}_i\|_2^2 + \gamma_i \|\underline{m}_i\|_2^2 \quad \text{for } i = 1, 2, \dots, 31. \quad (14)$$

Here, the per-channel regularization parameter γ_i (again to be optimized by a cross-validation procedure, see Section 2.2.1) is used specifically for regularizing the regression of the i th spectral channel. That is, we select different regularization parameters for different spectral channels.

We would like to remark that, although our per-channel approach matches the assumption made by the regression's formulation (that there is no inter-spectral-channel dependence), we shall admit the possibility that there might be better ways to formulate the regression which factors in “reasonable interdependence” between channels. For example, we may consider to impose a “smoothness” constraint used in the literature (see, e.g., in [76]) on the recovered spectra, though we note that this assumption would be more important for the recovery of “reflectance” spectra (which are intrinsically smooth) instead of the “radiance” spectra we are recovering (because the illumination spectrum is part of the radiance, which can occasionally make the radiance far from smooth, especially for indoor illuminations).

The solution of Equation (14) can be written in closed form [67]:

$$\underline{m}_i = [\mathbf{X}^T \mathbf{X} + \gamma_i \mathbf{I}]^{-1} \mathbf{X}^T \underline{\rho}_i. \quad (15)$$

Here, the $\mathbf{I}$ matrix is the identity matrix—if Linear Regression (Equation (5)) is used, then $\mathbf{I}$ is 3×3 ; otherwise, if a nonlinear transform is incorporated (Equation (7)), $\mathbf{I}$ is $s \times s$ where s is the dimension of the nonlinear feature vectors.

3.2. Relative-Error Least Squares Minimization

Now, let us minimize an error that is more similar to MRAE. First, we consider to formulate an ℓ_2 -minimization problem so as to ensure a closed-form solution.

From Equation (5), we can remodel the approximation as the following minimization:

$$\min_{\mathbf{M}} \left\| \frac{\mathbf{X}\mathbf{M} - \mathbf{R}}{\mathbf{R}} \right\|_2^2 = \min_{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \dots, \hat{\mathbf{r}}_N} \sum_{k=1}^N \left\| \frac{\hat{\mathbf{r}}_k - \mathbf{r}_k}{\mathbf{r}_k} \right\|_2^2, \quad \hat{\mathbf{r}}_k = \mathbf{M}^T \mathbf{x}_k, \quad (16)$$

where all the divisions are component-wise. Here, the square of an ℓ_2 relative error (referred to as the “relative-RMSE” in some works [55,70]) is minimized. We call this new minimization approach the Relative Error Least Squares (RELS).

Equation (16) can be rewritten in a per-channel fashion:

$$\min_{\hat{\rho}_1, \hat{\rho}_2, \dots, \hat{\rho}_{31}} \sum_{i=1}^{31} \left\| \frac{\hat{\rho}_i - \underline{\rho}_i}{\underline{\rho}_i} \right\|_2^2, \quad \hat{\rho}_i = \mathbf{X}\underline{m}_i \quad (17)$$

(these two minimizations are equivalent because the divisions are component-wise). Or equivalently, we write

$$\min_{\underline{m}_i} \left\| \frac{\mathbf{X}\underline{m}_i}{\underline{\rho}_i} - \underline{\mathbf{1}} \right\|_2^2 \quad \text{for } i = 1, 2, \dots, 31, \quad (18)$$

where $\underline{\mathbf{1}}$ is an N -component vector of ones.

We can further define a matrix of RGBs weighted by the reciprocals of $\underline{\rho}_i$:

$$\mathbf{H}_i = \frac{\mathbf{X}}{\underline{\rho}_i} = \begin{pmatrix} 1/\rho_{i,1} & 0 & \dots & 0 \\ 0 & 1/\rho_{i,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\rho_{i,N} \end{pmatrix} \mathbf{X}, \quad (19)$$

where $\underline{\rho}_i = [\rho_{i,1}, \rho_{i,2}, \dots, \rho_{i,N}]^T$. Using this nomenclature, let us again rewrite Equation (18) into

$$\min_{\underline{m}_i} \left\| \mathbf{H}_i \underline{m}_i - \underline{\mathbf{1}} \right\|_2^2 \quad \text{for } i = 1, 2, \dots, 31. \quad (20)$$

Clearly, Equation (20) shows that RELS is in effect another least-squares problem which regresses $\mathbf{H}_i$ to fit the vector $\underline{\mathbf{1}}$.

Of course, we need to regularize this minimization by solving the following equation instead:

$$\min_{\underline{m}_i} \left\| \mathbf{H}_i \underline{m}_i - \underline{\mathbf{1}} \right\|_2^2 + \gamma_i \left\| \underline{m}_i \right\|_2^2 \quad \text{for } i = 1, 2, \dots, 31, \quad (21)$$

whose closed-form solution is written as [67,77]

$$\underline{m}_i = [\mathbf{H}_i^T \mathbf{H}_i + \gamma_i \mathbf{I}]^{-1} \mathbf{H}_i^T \underline{\mathbf{1}}. \quad (22)$$

3.3. Relative-Error Least Absolute Deviation Minimization

Finally, let us consider to minimize MRAE directly. Analogous to Equation (16), we are now going to solve the following minimization:

$$\min_{\mathbf{M}} \left\| \frac{\mathbf{X}\mathbf{M} - \mathbf{R}}{\mathbf{R}} \right\|_1 = \min_{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \dots, \hat{\mathbf{r}}_N} \sum_{k=1}^N \left\| \frac{\hat{\mathbf{r}}_k - \mathbf{r}_k}{\mathbf{r}_k} \right\|_1, \quad \hat{\mathbf{r}}_k = \mathbf{M}^T \mathbf{x}_k. \quad (23)$$

Following the same derivation as Equation (16)–(21), we reach

$$\min_{\underline{m}_i} \left\| \mathbf{H}_i \underline{m}_i - \underline{\mathbf{1}} \right\|_1 + \gamma_i \left\| \underline{m}_i \right\|_1 \quad \text{for } i = 1, 2, \dots, 31. \quad (24)$$

In the literature, regressions solved via an ℓ_1 minimization is called the Least Absolute Deviation (LAD) [78,79]. As here the MRAE we are minimizing is a relative error, we call this new approach the Relative Error Least Absolute Deviation (RELAD).

Notice that for the regularization penalty term in Equation (24) we also adopted an ℓ_1 norm, which refers to the LASSO regularization [74] (see Section 2.2.1).

Unlike RELS, RELAD does not have a closed-form solution. For a small amount of data, a globally optimal solution can be found using a linear-programming solution [78,80]. However, in our application the amount of data is large—where the Iterative Reweighted Least Squares (IRLS) [79] algorithm is more appropriate and is thus used here.

The IRLS process approaches RELAD minimization by repeatedly solving Weighted Least Squares (WLS) [81] while updating the weights on every iteration depending on the losses and mapping functions obtained in the previous iteration, until the solution converges.

The detailed algorithm is given in Algorithm 1. The iteration number is indicated by the superscript (t) . All min, division, and absolute operations shown in Algorithm 1 are component-wise to the vectors, while the median function in Step 8 and the mean functions

in Step 11 calculate the median and mean of components of $\underline{\delta}^{(t)}$ (the resulting scalar $\hat{\sigma}$ in Step 8 is a preliminary estimate of the standard deviation of absolute losses commonly used in the literature, see, e.g., in [79,82]). Furthermore, the min functions used in Step 9 and 10 are to clip the reciprocal values at 10^6 so as to prevent overly large numbers. Finally, in Step 11, we set the tolerance $\epsilon = 0.00005$ and the stopping iteration $T = 20$.

Algorithm 1 Solving RELAD regression (Equation (24)) by IRLS algorithm.

```

1:  $\mathbf{W}^{(0)} = \widetilde{\mathbf{W}}^{(0)} = \mathbf{I}$  ▷ Initialization of weights;  $\mathbf{I}$  is the  $N \times N$  identity matrix
2:  $\underline{\delta}^{(0)} = \inf$  ▷ Absolute losses of all data are initialized to infinity
3:  $t = 0$ 
4: repeat
5:    $t = t + 1$ 
6:    $\underline{\mathbf{m}}_i^{(t)} = [\mathbf{H}_i^T \mathbf{W}^{(t-1)} \mathbf{H}_i + \gamma_i \widetilde{\mathbf{W}}^{(t-1)}]^{-1} \mathbf{H}_i^T \mathbf{W}^{(t-1)} \mathbf{1}$  ▷ Closed-form WLS solution
7:    $\underline{\delta}^{(t)} = |\mathbf{H}_i \underline{\mathbf{m}}_i^{(t)} - \mathbf{1}|$  ▷ Absolute losses
8:    $\hat{\sigma} = \frac{\text{median}(\underline{\delta}^{(t)})}{0.6745}$  ▷ Preliminary estimate of the standard deviation of  $\underline{\delta}^{(t)}$ 
9:    $\mathbf{W}^{(t)} = \hat{\sigma} \times \text{diag}\left(\min\left(\frac{1}{\underline{\delta}^{(t)}}, 10^6\right)\right)$  ▷ The  $\times$  operator is the scalar product
10:   $\widetilde{\mathbf{W}}^{(t)} = \text{diag}\left(\min\left(\frac{1}{|\underline{\mathbf{m}}_i^{(t)}|}, 10^6\right)\right)$ 
11: until  $|\text{mean}(\underline{\delta}^{(t)}) - \text{mean}(\underline{\delta}^{(t-1)})| < \epsilon$  or  $t \geq T$ 
12:  $\underline{\mathbf{m}}_i = \underline{\mathbf{m}}_i^{(t)}$  ▷ Return the converged  $\underline{\mathbf{m}}_i$ 

```

4. Experiment

4.1. The Regression Models

The regression models we consider in this paper (introduced in Section 2.1) are listed in Table 1. For each of these regressions, we train the model under the 4 minimization criteria listed in Table 2. That is, we benchmark 5 models paired with 4 minimization approaches; but, irrespective of which minimization we are using, we evaluate the performance of all algorithms using MRAE.

Table 1. List of considered regression models.

Model	Abbreviation
Linear Regression [52]	LR
Root-Polynomial Regression (6th order) [56]	RPR
A+ Sparse Coding [55]	A+
Radial Basis Function Network [54]	RBFN
Polynomial Regression (6th order) [53]	PR

Table 2. List of minimization approaches.

Approach	Abbreviation	Per-Channel Regularization	Loss Metric
Least Squares	LS	✗	squared RMSE
Per-Channel Least Squares	LS ^{pc}	✓	squared RMSE
Relative Error Least Squares	RELS	✓	squared relative-RMSE
Relative Error Least Absolute Deviation	RELAD	✓	MRAE

For reference, we benchmark against HSCNN-R [60]—the 2nd place winner of the NTIRE 2018 challenge [63]. HSCNN-R is DNN-based (orders of magnitude more parameters compared to all considered regressions), and it minimizes MRAE directly.

All implemented codes are provided as the Supplementary Materials.

4.2. Preparing Ground-Truth Datasets

In this paper, we adopt the ICVL hyperspectral image database [70], which was the database used for the NTIRE 2018 Spectral Reconstruction Challenge [63]. Each spectral image from the database (200 in total) has a spatial dimension of 1392×1300 —that is, approximately 1.8 M spectra per image—with 31 spectral channels (referring to 10-nm spectral samplings between 400 and 700 nm).

For each spectral image, we calculate the corresponding RGB image following Equation (2), using the CIE 1964 color matching functions [83,84] as the camera's spectral sensitivities S , i.e., the resulting RGBs are the CIEXYZ tristimulus values [84]. This setting corresponds to the “Clean Track” methodology of NTIRE 2018 and 2020 Spectral Reconstruction Challenge [63,64].

4.3. Training, Validation, and Testing

We follow a 4-trial cross-validation process. First, we randomly separated all images (pairs of hyperspectral and corresponding RGB images) into 4 data subsets—denoted as image subset $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$; each consists 50 scenes. Then, in each trial, 2 subsets were used for training, 1 subset was used for validation, and 1 subset was used for testing, and the roles for each subset in different trials were permuted as follows:

- Trial #1—Training: $\mathcal{A}$ & $\mathcal{B}$, Validation: $\mathcal{C}$, Testing: $\mathcal{D}$
- Trial #2—Training: $\mathcal{A}$ & $\mathcal{B}$, Validation: $\mathcal{D}$, Testing: $\mathcal{C}$
- Trial #3—Training: $\mathcal{C}$ & $\mathcal{D}$, Validation: $\mathcal{A}$, Testing: $\mathcal{B}$
- Trial #4—Training: $\mathcal{C}$ & $\mathcal{D}$, Validation: $\mathcal{B}$, Testing: $\mathcal{A}$.

In the training process of each trial, the training set was used to minimize for either the LS, LS^p , RELS, or RELAD criteria.

Then, the validation set was used to determine the unknown regularization parameters following the procedure introduced in Section 2.2.1. More specifically, we first coarsely tried a wide range of magnitudes: γ or $\gamma_i = \{10^{-20}, 10^{-19}, \dots, 10^{20}\}$, and then selected the magnitude that returns the lowest averaged validation-set MRAE. Next, we conducted a finer search at 1000 points (equidistant on the log scale) between the two adjacent magnitudes of the coarsely selected one, which determines the final selection of γ or γ_i .

For the deep learning-based HSCNN-R, the validation step is different. In HSCNN-R, we use the validation set to determine the termination epoch of the training [60]. The actual termination epochs adopted differ among the 4 cross-validation trials, but all of them are between 315 and 350 epochs.

Finally, we evaluated all models and minimization criteria using the testing-set images. The resulting statistics presented in the next section are the averaged testing results over the 4 cross-validation trials.

5. Results

5.1. Mean and Worst-Case Performance

In Table 3, we present the mean statistics of per-image-mean and per-image-99-percentile (worst-case) MRAE over the 200 tested images. Let us first look at the numbers in the first row, which are results of variations of LR (i.e., Linear Regression). The mean results (under the headline “Mean per-image-mean MRAE”) show that LR trained using all of our three new approaches outperform the standard LS (least-squares), among which the RELAD method performs the best—returning 14% lower MRAE compared to LS. On the right side of Table 3 (headlined “Mean per-image-99-percentile MRAE”), we see that RELS-based LR provides 17% lower worst-case MRAE compared to using the standard LS. Likewise, observing all other regression models, we found that the best minimization criterion in terms of mean MRAE is either RELS or RELAD depending on the model.

To further examine the robustness of the best minimization approach against other tested approaches, we present the “paired” (or “dependent”) two-sample Student's t -test

scores [85] in Table 4. The t -test scores are calculated based on the observed 200 pairs of per-image-mean MRAEs delivered by two compared minimization criteria. For example, let us look at the top-left number of the table, where 9.46 is the t -test score when comparing the “Best” approach for LR (i.e., RELAD, which delivers the lowest mean per-image-mean MRAE for LR) versus the conventional LS minimization. In the same row, we also see the t -test scores comparing the Best approach with LSP^c and RELS, respectively, but not RELAD (because RELAD is the Best criterion itself).

Table 3. The mean per-image-mean and per-image-99-percentile MRAE for all tested spectral reconstruction models. The result of HSCNN-R (the reference DNN model) is given in the last row. For all regression approaches (all except HSCNN-R), best results among the 4 training criteria—LS, LSP^c, RELS, and RELAD—are underlined.

	Mean Per-Image-Mean MRAE				Mean Per-Image-99-Percentile MRAE			
	LS	LSP ^c	RELS	RELAD	LS	LSP ^c	RELS	RELAD
LR	0.0624	0.0605	0.0563	<u>0.0536</u>	0.1695	0.1779	<u>0.1409</u>	0.1656
RPR	0.0469	0.0431	<u>0.0418</u>	0.0471	0.1548	0.1498	<u>0.1294</u>	0.1432
A+	0.0387	0.0375	0.0360	<u>0.0349</u>	0.1526	0.1428	<u>0.1350</u>	0.1402
RBFN	0.0206	0.0203	<u>0.0198</u>	0.0203	0.0789	0.0793	<u>0.0740</u>	0.0799
PR	0.0195	0.0193	<u>0.0188</u>	0.0192	0.0710	0.0709	<u>0.0703</u>	0.0734
HSCNN-R			0.0173				0.0653	

Table 4. The paired two-sample Student’s t -test scores of the mean per-image-mean MRAE results. For each regression model, the best minimization criterion (the one has the lowest number in Table 3) is tested against each of the other criteria.

	Best Approach	Student’s t -Test Score of Mean Per-Image-Mean MRAE			
		Best vs. LS	Best vs. LSP ^c	Best vs. RELS	Best vs. RELAD
LR	RELAD	9.46	9.54	4.18	N/A
RPR	RELS	7.70	2.65	N/A	13.28
A+	RELAD	6.93	7.19	7.69	N/A
RBFN	RELS	4.42	3.01	N/A	5.33
PR	RELS	4.59	4.18	N/A	4.33

In essence, the larger the absolute value of the t -test score, the more significant the two distributions are apart (i.e., the advantageous mean performance of the Best criterion has statistical significance). In our case, all tests have a degree of freedom of $200 - 1 = 199$ (200 is the number of dependent sample pairs). Accordingly, with a “one-sided” hypothesis (as we only want to test if the mean of the Best criterion is smaller), a 5% level of significance (so-called the “ p -value” in most literature) corresponds to a t -score of 1.65 (according to the t -distribution table in [86]). Evidently, we see that the t -scores of all our tests are greater than the 1.65 threshold. In a practical sense, this result suggests that for each regression model, using the corresponding best minimization criterion can consistently deliver the lowest per-image-mean MRAEs of all.

Still, it is somewhat counterintuitive that RELAD performs worse than RELS in some cases. Indeed, as RELAD directly minimizes MRAE, we might expect that RELAD would always provide the lowest mean MRAE. This outcome likely originates from the imprecision problem of IRLS specifically when solving LAD (i.e., the ℓ_1 Least Absolute Deviation) minimizations [79,87,88]. Another possible cause which might result in suboptimal optimization is that the IRLS algorithm might, in some cases, fail to converge after 20 iterations (remember that we set the stopping iteration $T = 20$ in Step 11 of Algorithm 1). We stopped iterating after 20 iterations to keep limit the computational complexity of the problem.

In contrast, it is not entirely surprising that RELS has better worst-case performance than RELAD (and consequently RELS is the best method for all tested regressions in terms of the worst-case MRAE). Indeed, it is well known that ℓ_1 minimizations tend to be less

sensitive to outliers compared to their ℓ_2 counterparts [78,79]. Which means, in RELAD these 99-percentile pixels might be treated as outliers during training—i.e., minimization of these pixels are less important—and thus perform worse in testing.

Finally, it is shown that our best performing regression model—the RELS-based PR—is only 8.7% worse than HSCNN-R in terms of mean MRAE. We remind the reader that HSCNN-R is one of the top models in the NTIRE 2018 challenge [63] in which all finalists are based on DNNs. Given that most of the reported challenge entries were much more than 10% worse than HSCNN-R under the MRAE evaluation, we can expect that some of our further optimized regression models can be on par with—or even better than—many DNN models in the challenge.

This result contradicts the assumption taken by many DNN approaches that mapping large image patches is necessary for achieving top performances—as all tested regressions in our experiment are pixel-based. Furthermore, regarding how much fewer model parameters the regressions use in comparison to the DNNs (e.g., the PR regression model has 2573 parameters, whereas HSCNN-R has approximately 10^7 parameters), it is surprising to see that regression methods can achieve comparable performance to the DNNs (let alone being better than some). As DNNs have millions of parameters, it is likely that the training data is insufficient to robustly optimize these parameters (the ICVL database [70] used in our study and in NTIRE 2018 challenge [63] is already one of the largest available hyperspectral image databases so far).

5.2. Computational Time

The training and reconstruction time of all tested models are presented in Table 5. We calculate the training time as the average time used for each cross-validation trial, and the reconstruction time as the average time used to reconstruct each tested image.

Table 5. The training and reconstruction time measurements. The size of the reconstructed images (from the ICVL database) is 1392×1300 with 31 spectral channels, and the training set (for each cross-validation trial) is composed of 100 images.

	Training Time (Per Cross-Validation Trial)				Reconstruction Time (Per Image)			
	LS	LSP ^c	RELS	RELAD	LS	LSP ^c	RELS	RELAD
LR	6.7 min	6.1 min	6.2 min	36.0 h	3.3 s	3.3 s	3.3 s	3.3 s
RPR	17.0 min	16.9 min	35.9 min	47.8 h	10.8 s	10.8 s	10.8 s	10.9 s
A+	26.9 min	52.8 min	54.6 min	46.2 h	1.5 min	1.5 min	1.5 min	1.5 min
RBFN	1.0 h	1.0 h	1.2 h	36.8 h	7.0 s	7.0 s	7.0 s	7.0 s
PR	15.1 min	15.4 min	42.2 min	44.6 h	9.8 s	10.0 s	10.0 s	9.8 s
HSCNN-R	35.5 h (GPU-accelerated)				3.4 min			

Our hardware specification includes Intel® Core™ i7-9700 CPU and NVIDIA® GeForce® RTX 2080 SUPER™ GPU. Note that the GPU is only used to run the training process of the HSCNN-R model. All testing (reconstruction) processes and the training of the regressions use solely the CPU.

Let us look at the training time (the left side of Table 5). First, we see that regardless of the regression model, the closed-form LS, LSP^c and RELS approaches show absolute dominance over the iteratively solved RELAD approach and HSCNN-R for shorter training time. Then, although it appears that the RELAD minimization in general takes longer than training HSCNN-R, we remark that HSCNN-R is GPU-accelerated (which is highly based on parallel programming), yet our implementation of RELAD does not use any apparent parallel programming technique.

As for the reconstruction time results (the right side of Table 5), we see that all tested regression methods require much less time to reconstruct a spectral image compared to HSCNN-R. Additionally, we notice that the reconstruction time for regression models does not depend on the adopted minimization approach—this result makes sense because after

all, no matter how we optimized the regression matrix, the reconstruction procedure is the same. Consequently, in the case that longer “offline” training time is permitted, the RELAD-optimized regression can still support fast reconstruction as LS, LS^{PC} , and RELS.

5.3. Brightness Dependence

Let us further investigate the performance discrepancy within an image. In Figure 4, we show for each regression approach the MRAE error map of a given example scene. We see that RELS and RELAD tend to improve the spectral recoveries for the foreground objects particularly e.g., trees and grass in the LR results, flowers in the RPR results, leaves in the A+ results, bonsai pot in the RBFN results, and the green peppers in the PR results. Yet, it seems that in the background and/or highlight regions (e.g., sandy grounds and the blue sky), LS and LS^{PC} in turn outperform RELS and RELAD.

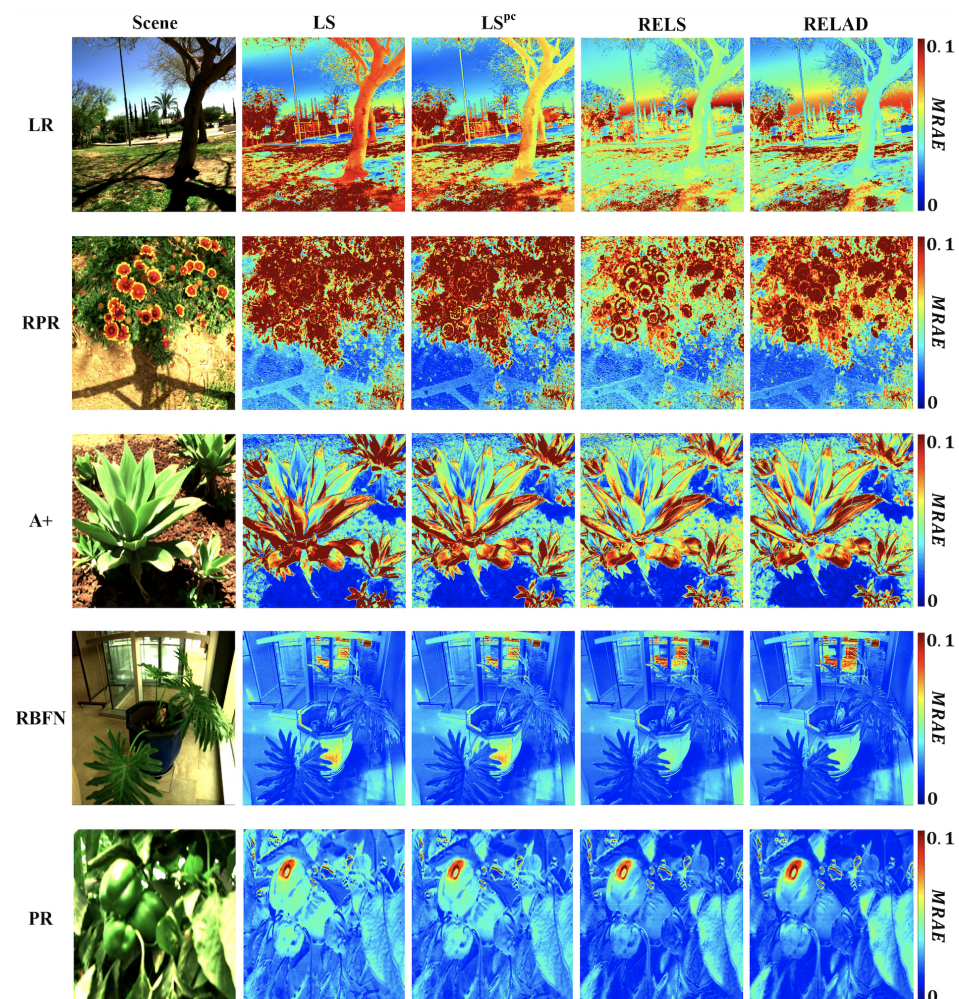


Figure 4. The MRAE error heat maps for the considered regression models optimized for LS, LS^{PC} , RELS, and RELAD criteria.

We observe that generally in scenes included in the ICVL database the foregrounds are dimmer than the backgrounds, therefore we presume the reason that LS and LS^{PC} tend to recover spectra of the background and highlights more accurately (yet perform much worse in the foreground) is due to the brightness bias of least-squares minimization (see Section 2.3).

In Figure 5, we exhibit how the mean MRAE performance of each regression approach is related to the pixels’ brightness. Here, “brightness” is defined as the ℓ_2 norm of the ground-truth spectrum. For each image we separated all pixels into 4 different brightness groups—0–25 percentile (the dimmest group), 25–50 percentile, 50–75 percentile, and

75–100 percentile (the brightest group). Then, for pixels belonging to the same brightness group (from all testing-set images) we calculated their mean MRAE, which is presented in the plots.

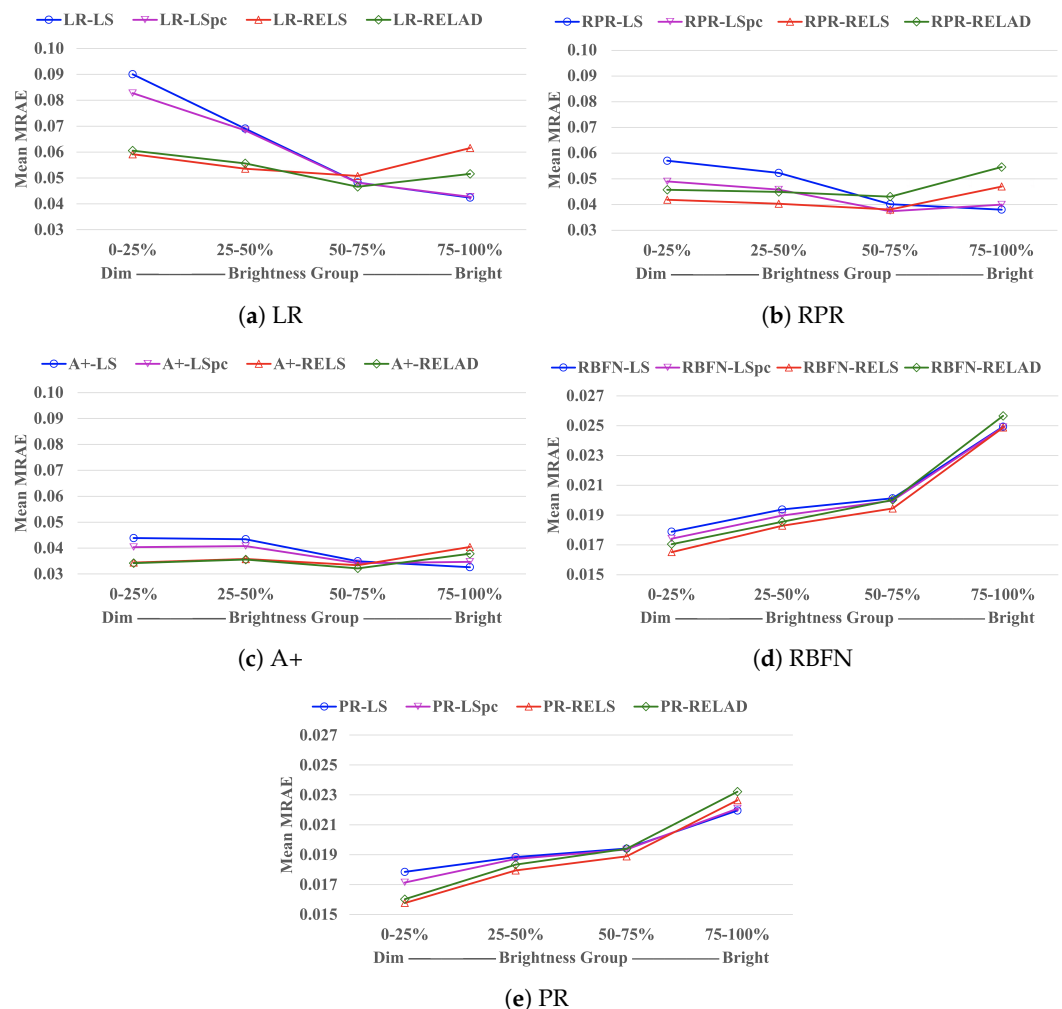


Figure 5. The mean MRAE performance for pixels belonging to 4 different percentile groups (based on their brightness) in each image. For display purposes, in plot (a–c) the Mean MRAE (the vertical axes) is shown between the interval of [0.03, 0.10], while in plot (d,e) it is shown between [0.015, 0.027].

Clearly, we observe that while LS and LS^{pc} generally provide lower MRAE for the brightest group in an image (75–100%), RELS and RELAD demonstrate great advantages over LS and LS^{pc} in the two dimmest groups (0–25% and 25–50%), which eventually leads to better overall performance.

6. Conclusions

Spectral reconstruction (SR) recovers high-resolution radiance spectra from RGB images. Many methods are regression-based, with simple formulations and usually closed-form solutions, while the current state-of-the-art spectral recovery is delivered by the much more sophisticated Deep Neural Network (DNN) solutions.

Recently, the top DNN models are trained and evaluated based on Mean Relative Absolute Error (MRAE)—a relative error which measures the spectral difference as a percentage of the ground-truth spectral value. Comparatively, all regressions are still trained based on the Least Squares (LS) minimization, which does not suggest a minimized MRAE result. This problem is further compounded by the sub-optimal regularization setting used in conventional regressions where all spectral channels are jointly regularized by a single penalty parameter.

In this paper, we developed new regression approaches that minimize relative errors and are regularized per spectral channel, including the closed-form Relative-Error Least Squares (RELS) and the Relative-Error Least Absolute Deviation (RELAD) approach (which directly minimizes MRAE and was solved by an iterative method).

Our results showed that the new minimization approaches significantly improve the conventional regressions especially in the darker regions of the images. Consequently, our best improved regression model narrows the performance gap with the leading DNNs to only 8% under the MRAE evaluation.

Supplementary Materials: The code of the methods introduced in this paper is available at https://github.com/EthanLinYitun/On_the_Optimization_of_Regression_Based_Spectral_Reconstruction.

Author Contributions: Conceptualization, Y.-T.L. and G.D.F.; Data curation, Y.-T.L.; Formal analysis, Y.-T.L.; Funding acquisition, G.D.F.; Investigation, Y.-T.L.; Methodology, Y.-T.L. and G.D.F.; Project administration, G.D.F.; Resources, G.D.F.; Software, Y.-T.L.; Supervision, G.D.F.; Validation, Y.-T.L.; Visualization, Y.-T.L.; Writing—original draft, Y.-T.L.; Writing—review and editing, Y.-T.L. and G.D.F. Both authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by EPSRC grant EP/S028730/1. We also thank Apple Inc. for additional support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <http://icvl.cs.bgu.ac.il/hyperspectral/>.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hardeberg, J.Y. On the spectral dimensionality of object colours. In Proceedings of the Conference on Colour in Graphics, Imaging, and Vision, Poitiers, France, 2–5 April 2002; Society for Imaging Science and Technology: Scottsdale, AZ, USA, 2002; Volume 2002, pp. 480–485.
2. Romero, J.; Garcia-Beltrán, A.; Hernández-Andrés, J. Linear bases for representation of natural and artificial illuminants. *J. Opt. Soc. Am. A* **1997**, *14*, 1007–1014.
3. Lee, T.W.; Wachtler, T.; Sejnowski, T.J. The spectral independent components of natural scenes. In Proceedings of the International Workshop on Biologically Motivated Computer Vision, Seoul, Korea, 15–17 May 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 527–534.
4. Marimont, D.H.; Wandell, B.A. Linear models of surface and illuminant spectra. *J. Opt. Soc. Am. A* **1992**, *9*, 1905–1913.
5. Parkkinen, J.P.; Hallikainen, J.; Jaaskelainen, T. Characteristic spectra of Munsell colors. *J. Opt. Soc. Am. A* **1989**, *6*, 318–322.
6. Riihihaho, K.A.; Eskelinen, M.A.; Pölönen, I. A Do-It-Yourself Hyperspectral Imager Brought to Practice with Open-Source Python. *Sensors* **2021**, *21*, 1072.
7. Stuart, M.B.; McGonigle, A.J.; Davies, M.; Hobbs, M.J.; Boone, N.A.; Stanger, L.R.; Zhu, C.; Perin, T.D.; Willmott, J.R. Low-Cost Hyperspectral Imaging with A Smartphone. *J. Imaging* **2021**, *7*, 136.
8. Zhao, Y.; Guo, H.; Ma, Z.; Cao, X.; Yue, T.; Hu, X. Hyperspectral Imaging With Random Printed Mask. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 10149–10157.
9. Garcia, H.; Correa, C.V.; Arguello, H. Multi-resolution compressive spectral imaging reconstruction from single pixel measurements. *IEEE Trans. Image Process.* **2018**, *27*, 6174–6184.
10. Takatani, T.; Aoto, T.; Mukaigawa, Y. One-shot hyperspectral imaging using faced reflectors. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4039–4047.
11. Galvis, L.; Lau, D.; Ma, X.; Arguello, H.; Arce, G.R. Coded aperture design in compressive spectral imaging based on side information. *Appl. Opt.* **2017**, *56*, 6332–6340.
12. Wang, L.; Xiong, Z.; Gao, D.; Shi, G.; Zeng, W.; Wu, F. High-speed hyperspectral video acquisition with a dual-camera architecture. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4942–4950.
13. Rueda, H.; Arguello, H.; Arce, G.R. DMD-based implementation of patterned optical filter arrays for compressive spectral imaging. *J. Opt. Soc. Am. A* **2015**, *32*, 80–89.
14. Correa, C.V.; Arguello, H.; Arce, G.R. Snapshot colored compressive spectral imager. *J. Opt. Soc. Am. A* **2015**, *32*, 1754–1763.
15. Arguello, H.; Arce, G.R. Colored coded aperture design by concentration of measure in compressive spectral imaging. *IEEE Trans. Image Process.* **2014**, *23*, 1896–1908.
16. Lin, X.; Liu, Y.; Wu, J.; Dai, Q. Spatial-spectral encoded compressive hyperspectral imaging. *ACM Trans. Graph.* **2014**, *33*, 233.

17. Cao, X.; Du, H.; Tong, X.; Dai, Q.; Lin, S. A prism-mask system for multispectral video acquisition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2423–2435.
18. Gat, N. Imaging spectroscopy using tunable filters: A review. In Proceedings of the Wavelet Applications VII, Orlando, FL, USA, 26–28 April 2000; International Society for Optics and Photonics: Bellingham, WA, USA, 2000; Volume 4056, pp. 50–64.
19. Green, R.O.; Eastwood, M.L.; Sarture, C.M.; Chrien, T.G.; Aronsson, M.; Chippendale, B.J.; Faust, J.A.; Pavri, B.E.; Chovit, C.J.; Solis, M.; et al. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sens. Environ.* **1998**, *65*, 227–248.
20. Wang, W.; Ma, L.; Chen, M.; Du, Q. Joint Correlation Alignment-Based Graph Neural Network for Domain Adaptation of Multitemporal Hyperspectral Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3170–3184.
21. Torun, O.; Yuksel, S.E. Unsupervised segmentation of LiDAR fused hyperspectral imagery using pointwise mutual information. *Int. J. Remote Sens.* **2021**, *42*, 6465–6480.
22. Tu, B.; Zhou, C.; Liao, X.; Zhang, G.; Peng, Y. Spectral-spatial hyperspectral classification via structural-kernel collaborative representation. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 861–865.
23. Inamdar, D.; Kalacska, M.; Leblanc, G.; Arroyo-Mora, J.P. Characterizing and mitigating sensor generated spatial correlations in airborne hyperspectral imaging data. *Remote Sens.* **2020**, *12*, 641.
24. Alcolea, A.; Paoletti, M.E.; Haut, J.M.; Resano, J.; Plaza, A. Inference in supervised spectral classifiers for on-board hyperspectral imaging: An overview. *Remote Sens.* **2020**, *12*, 534.
25. Gholizadeh, H.; Gamon, J.A.; Helzer, C.J.; Cavender-Bares, J. Multi-temporal assessment of grassland α - and β -diversity using hyperspectral imaging. *Ecol. Appl.* **2020**, *30*, e02145.
26. Veganzones, M.; Tochon, G.; Dalla-Mura, M.; Plaza, A.; Chanussot, J. Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation. *IEEE Trans. Image Process.* **2014**, *23*, 3574–3589.
27. Ghamisi, P.; Dalla Mura, M.; Benediktsson, J. A survey on spectral-spatial classification techniques based on attribute profiles. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 2335–2353.
28. Lv, M.; Chen, T.; Yang, Y.; Tu, T.; Zhang, N.; Li, W.; Li, W. Membranous nephropathy classification using microscopic hyperspectral imaging and tensor patch-based discriminative linear regression. *Biomed. Opt. Express* **2021**, *12*, 2968–2978.
29. Courtenay, L.A.; González-Aguilera, D.; Lagüela, S.; del Pozo, S.; Ruiz-Mendez, C.; Barbero-García, I.; Román-Curto, C.; Cañueto, J.; Santos-Durán, C.; Cardeñoso-Álvarez, M.E.; et al. Hyperspectral imaging and robust statistics in non-melanoma skin cancer analysis. *Biomed. Opt. Express* **2021**, *12*, 5107–5127.
30. Zhang, Y.; Xi, Y.; Yang, Q.; Cong, W.; Zhou, J.; Wang, G. Spectral CT reconstruction with image sparsity and spectral mean. *IEEE Trans. Comput. Imaging* **2016**, *2*, 510–523.
31. Zhang, Y.; Mou, X.; Wang, G.; Yu, H. Tensor-based dictionary learning for spectral CT reconstruction. *IEEE Trans. Med Imaging* **2016**, *36*, 142–154.
32. Chen, Z.; Wang, J.; Wang, T.; Song, Z.; Li, Y.; Huang, Y.; Wang, L.; Jin, J. Automated in-field leaf-level hyperspectral imaging of corn plants using a Cartesian robotic platform. *Comput. Electron. Agric.* **2021**, *183*, 105996.
33. Gomes, V.; Mendes-Ferreira, A.; Melo-Pinto, P. Application of Hyperspectral Imaging and Deep Learning for Robust Prediction of Sugar and pH Levels in Wine Grape Berries. *Sensors* **2021**, *21*, 3459.
34. Pane, C.; Manganiello, G.; Nicastro, N.; Cardi, T.; Carotenuto, F. Powdery Mildew Caused by *Erysiphe cruciferarum* on Wild Rocket (*Diplotaxis tenuifolia*): Hyperspectral Imaging and Machine Learning Modeling for Non-Destructive Disease Detection. *Agriculture* **2021**, *11*, 337.
35. Hu, N.; Li, W.; Du, C.; Zhang, Z.; Gao, Y.; Sun, Z.; Yang, L.; Yu, K.; Zhang, Y.; Wang, Z. Predicting micronutrients of wheat using hyperspectral imaging. *Food Chem.* **2021**, *343*, 128473.
36. Weksler, S.; Rozenstein, O.; Haish, N.; Moshelion, M.; Wallach, R.; Ben-Dor, E. Detection of Potassium Deficiency and Momentary Transpiration Rate Estimation at Early Growth Stages Using Proximal Hyperspectral Imaging and Extreme Gradient Boosting. *Sensors* **2021**, *21*, 958.
37. Qin, J.; Chao, K.; Kim, M.S.; Lu, R.; Burks, T.F. Hyperspectral and multispectral imaging for evaluating food safety and quality. *J. Food Eng.* **2013**, *118*, 157–171.
38. Xie, W.; Fan, S.; Qu, J.; Wu, X.; Lu, Y.; Du, Q. Spectral Distribution-Aware Estimation Network for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, doi:10.1109/TGRS.2021.3089711.
39. Zhang, L.; Cheng, B. A combined model based on stacked autoencoders and fractional Fourier entropy for hyperspectral anomaly detection. *Int. J. Remote Sens.* **2021**, *42*, 3611–3632.
40. Li, X.; Zhao, C.; Yang, Y. Hyperspectral anomaly detection based on the distinguishing features of a redundant difference-value network. *Int. J. Remote Sens.* **2021**, *42*, 5459–5477.
41. Zhang, X.; Ma, X.; Huan, N.; Gu, J.; Tang, X.; Jiao, L. Spectral-Difference Low-Rank Representation Learning for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, doi:10.1109/TGRS.2020.3046727.
42. Yang, Y.; Song, S.; Liu, D.; Chan, J.C.W.; Li, J.; Zhang, J. Hyperspectral anomaly detection through sparse representation with tensor decomposition-based dictionary construction and adaptive weighting. *IEEE Access* **2020**, *8*, 72121–72137.
43. Lei, J.; Fang, S.; Xie, W.; Li, Y.; Chang, C.I. Discriminative reconstruction for hyperspectral anomaly detection with spectral learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7406–7417.

44. Jablonski, J.A.; Bihl, T.J.; Bauer, K.W. Principal component reconstruction error for hyperspectral anomaly detection. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1725–1729.
45. Cheung, V.; Westland, S.; Li, C.; Hardeberg, J.; Connah, D. Characterization of trichromatic color cameras by using a new multispectral imaging technique. *J. Opt. Soc. Am. A* **2005**, *22*, 1231–1240.
46. Shen, H.L.; Xin, J.H. Spectral characterization of a color scanner by adaptive estimation. *J. Opt. Soc. Am. A* **2004**, *21*, 1125–1130.
47. Jaspe-Villanueva, A.; Ahsan, M.; Pintus, R.; Giachetti, A.; Marton, F.; Gobbetti, E. Web-based Exploration of Annotated Multi-Layered Relightable Image Models. *ACM J. Comput. Cult. Herit.* **2021**, *14*, 1–29.
48. Lam, A.; Sato, I. Spectral modeling and relighting of reflective-fluorescent scenes. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1452–1459.
49. Picollo, M.; Cucci, C.; Casini, A.; Stefani, L. Hyper-spectral imaging technique in the cultural heritage field: New possible scenarios. *Sensors* **2020**, *20*, 2843.
50. Grillini, F.; Thomas, J.B.; George, S. Mixing models in close-range spectral imaging for pigment mapping in cultural heritage. In Proceedings of the International Colour Association (AIC) Conference, Online, 20 & 26–27 November 2020; pp. 372–376.
51. Xu, P.; Xu, H.; Diao, C.; Ye, Z. Self-training-based spectral image reconstruction for art paintings with multispectral imaging. *Appl. Opt.* **2017**, *56*, 8461–8470.
52. Heikkinen, V.; Lenz, R.; Jetsu, T.; Parkkinen, J.; Hauta-Kasari, M.; Jääskeläinen, T. Evaluation and unification of some methods for estimating reflectance spectra from RGB images. *J. Opt. Soc. Am. A* **2008**, *25*, 2444–2458.
53. Connah, D.; Hardeberg, J. Spectral recovery using polynomial models. Color Imaging X: Processing, Hardcopy, and Applications in Proceedings of the Electronic Imaging, San Jose, CA, USA, 16–20 January 2005; International Society for Optics and Photonics: Bellingham, WA, USA, 2005; Volume 5667, pp. 65–75.
54. Nguyen, R.; Prasad, D.; Brown, M. Training-based spectral reconstruction from a single RGB image. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 186–201.
55. Aeschbacher, J.; Wu, J.; Timofte, R. In defense of shallow learned spectral reconstruction from RGB images. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 471–479.
56. Lin, Y.T.; Finlayson, G.D. Exposure Invariance in Spectral Reconstruction from RGB Images. In Proceedings of the Color and Imaging Conference, Paris, France, 21–25 October 2019; Society for Imaging Science and Technology: Scottsdale, AZ, USA, 2019; Volume 2019; pp. 284–289.
57. Lin, Y.T.; Finlayson, G.D. Physically Plausible Spectral Reconstruction. *Sensors* **2020**, *20*, 6399.
58. Stiebel, T.; Merhof, D. Brightness Invariant Deep Spectral Super-Resolution. *Sensors* **2020**, *20*, 5789.
59. Lin, Y.T. Colour Fidelity in Spectral Reconstruction from RGB Images. In Proceedings of the London Imaging Meeting, London, UK, 29 September–1 October 2020; Society for Imaging Science and Technology: Scottsdale, AZ, USA, 2020; Volume 2020, pp. 144–148.
60. Shi, Z.; Chen, C.; Xiong, Z.; Liu, D.; Wu, F. Hscnn+: Advanced cnn-based hyperspectral recovery from RGB images. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 939–947.
61. Li, J.; Wu, C.; Song, R.; Li, Y.; Liu, F. Adaptive weighted attention network with camera spectral sensitivity prior for spectral reconstruction from RGB images. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 462–463.
62. Zhao, Y.; Po, L.M.; Yan, Q.; Liu, W.; Lin, T. Hierarchical regression network for spectral reconstruction from RGB images. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 422–423.
63. Arad, B.; Ben-Shahar, O.; Timofte, R. NTIRE 2018 challenge on spectral reconstruction from RGB images. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 929–938.
64. Arad, B.; Timofte, R.; Ben-Shahar, O.; Lin, Y.T.; Finlayson, G.D. NTIRE 2020 challenge on spectral reconstruction from an RGB Image. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020.
65. Arun, P.; Buddhiraju, K.; Porwal, A.; Chanussot, J. CNN based spectral super-resolution of remote sensing images. *Signal Process.* **2020**, *169*, 107394.
66. Joslyn Fubara, B.; Sedky, M.; Dyke, D. RGB to Spectral Reconstruction via Learned Basis Functions and Weights. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 480–481.
67. Tikhonov, A.; Goncharsky, A.; Stepanov, V.; Yagola, A. *Numerical Methods for the Solution of Ill-Posed Problems*; Springer Science & Business Media: New York, NY, USA, 2013; Volume 328.
68. Webb, G.I. Overfitting. In *Encyclopedia of Machine Learning*; Sammut, C., Webb, G.I., Eds.; Springer: Boston, MA, USA, 2010; pp. 744–744.
69. Wandell, B.A. The synthesis and analysis of color images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9*, 2–13.
70. Arad, B.; Ben-Shahar, O. Sparse recovery of hyperspectral signal from natural RGB images. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 19–34.

71. Aharon, M.; Elad, M.; Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **2006**, *54*, 4311–4322.
72. Penrose, R. A generalized inverse for matrices. In *Mathematical Proceedings of the Cambridge Philosophical Society*; Cambridge University Press: Cambridge, UK, 1955; Volume 51, pp. 406–413.
73. Galatsanos, N.; Katsaggelos, A. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Trans. Image Process.* **1992**, *1*, 322–336.
74. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **1996**, *58*, 267–288.
75. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2005**, *67*, 301–320.
76. Van Trigt, C. Smoothest reflectance functions. II. Complete results. *J. Opt. Soc. Am. A* **1990**, *7*, 2208–2222.
77. Tofallis, C. Least squares percentage regression. *J. Mod. Appl. Stat. Methods* **2008**, *7*, 526–534.
78. Wang, L.; Gordon, M.D.; Zhu, J. Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In Proceedings of the International Conference on Data Mining, Hong Kong, China, 18–22 December 2006; pp. 690–700.
79. Chen, K.; Lv, Q.; Lu, Y.; Dou, Y. Robust regularized extreme learning machine for regression using iteratively reweighted least squares. *Neurocomputing* **2017**, *230*, 345–358.
80. Wagner, H.M. Linear programming techniques for regression analysis. *J. Am. Stat. Assoc.* **1959**, *54*, 206–212.
81. Carroll, R.J.; Ruppert, D. *Transformation and Weighting in Regression*; CRC Press: Boca Raton, FL, USA, 1988; Volume 30.
82. Deng, W.; Zheng, Q.; Chen, L. Regularized extreme learning machine. In Proceedings of the Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA, 30 March–2 April 2009; pp. 389–395.
83. Commission Internationale de L’éclairage. *CIE Proceedings (1964) Vienna Session, Committee Report E-1.4. 1*; Commission Internationale de L’éclairage: Paris, France, 1964.
84. Wyszecki, G.; Stiles, W.S. *Color Science*; Wiley: New York, NY, USA, 1982; Volume 8.
85. Snedecor, G.W.; Cochran, W. *Statistical Methods*, 6th ed.; The Iowa State University: Ames, IA, USA, 1967; pp. 91–119.
86. Kokoska, S.; Zwillinger, D. *CRC Standard Probability and Statistics Tables and Formulae*; CRC Press: Boca Raton, FL, USA, 2000; pp. 152–153.
87. Schlossmacher, E. An iterative technique for absolute deviations curve fitting. *J. Am. Stat. Assoc.* **1973**, *68*, 857–859.
88. Gentle, J.E. *Matrix Algebra*; Springer Texts in Statistics; Springer: Cham, Switzerland, 2007; Volume 10, pp. 232–233.