Per-Channel Regularization for Regression-Based Spectral Reconstruction

Yi-Tun Lin and Graham D. Finlayson

School of Computing Sciences, University of East Anglia, Norwich, UK
Yi-Tun.Lin@uea.ac.uk, G.Finlayson@uea.ac.uk

Abstract. Spectral reconstruction algorithms seek to recover spectra from RGB images. This estimation problem is often formulated as *least-squares* regression, and a *Tikhonov regularization* is generally incorporated, both to support stable estimation in the presence of noise and to prevent over-fitting. The degree of regularization is controlled by a single penalty-term parameter, which is often selected using the *cross validation* experimental methodology. In this paper, we generalize the simple regularization approach to admit a per-spectral-channel optimization setting, and a modified cross-validation procedure is developed. Experiments validate our method. Compared to the conventional regularization, our per-channel approach significantly improves the reconstruction accuracy at multiple spectral channels, by up to 17% increments for all the considered models.

Keywords: Spectral reconstruction · Hyperspectral imaging · Multi-spectral imaging.

1 Introduction

The light spectrum is a continuous intensity distribution across wavelengths. This *spectral information* is commonly used to help determine and/or discriminate the physical properties of object surfaces, for example in remote sensing [25, 8, 14, 23] and medical imaging [28, 29]. Also, in various practical applications, the devices (sensors or displays), light sources and object surfaces are characterized by spectral measurements [1, 9, 16, 27].

Despite the advantages of spectral capture, almost all images that we record contain just 3 measurements - the 3 weighted spectral averages over the **R**ed, **G**reen and **B**lue spectral regions. Perforce, much spectral information is therefore *lost* in this RGB image formation process. Indeed, it is a classical result in color science, that there are many spectra - called *metamers* [12] - which integrate to the same RGB, and of course, given only one RGB measurement we cannot tell which physical spectrum induced it. Still, by adopting learning approaches we can estimate the spectrum that *likely* corresponds to a given RGB. Estimating spectra from RGBs is called *spectral reconstruction* (SR).

Colour and Visual Computing Symposium 2020, Gjøvik, Norway, September 16-17, 2020.

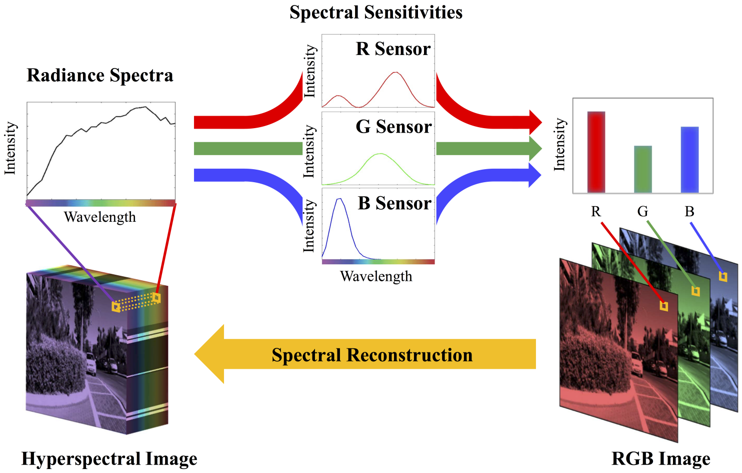

Fig. 1. RGB image formation (Hyperspectral → RGB image) and spectral reconstruction (RGB → Hyperspectral image). For illustration we color-coded the wavelength scale by the colors we would see when observing each single wavelength of light.

In Fig. 1 we illustrate RGB image formation and the SR process. In the top-left panel we see a single radiance spectrum measured at one location in the hyperspectral image (bottom left). This spectrum is sampled by 3 sensors, resulting in the 3-value RGB response (top-right). Repeating this process for all image locations, the corresponding RGB image in the bottom right is derived from the hyperspectral image. Then, the spectral reconstruction algorithms attempt to recover the hyperspectral image back from the RGB image (or an *approximation* thereof).

Historically, this SR problem is effectively solved by *least-squares regression* [15, 11, 17, 18, 2], where the map from RGBs (or the non-linear RGB features) to spectra is modelled as a simple linear transform. More recently, deep learning approaches [4, 22, 6] have been developed that provide even better SR performance. Effectively, this performance increment is achieved by regressing an RGB in the context of its *neighborhood* to its corresponding spectra. Clearly, this patch-based idea has merit. For example, if the algorithm can identify a patch in the scene as a 'skin region' then spectral recovery is plausibly easier to solve, since skin spectra have characteristic spectral shapes [7, 19].

Despite clear rationale behind the deep-learning approach, Aeschbacher *et al.* [2] show that the regression-based A+ algorithm provides very competitive performance. Moreover, Lin and Finlayson [17] shows that several regression methods actually generalize better than the leading deep-learning models when the scene exposure changes.

The main concern of this paper is in the *regularization* step of regression-based SR algorithms. The classical (multivariate) regression problem from statistics is written as

$$\mathbf{MA} \approx \mathbf{B} \ , \tag{1}$$

where $\mathbf{A}$ is an $m \times N$ matrix as the table of measured data (m is the dimension of the measured data and N is the number of data samples, with $N \gg m$), and $\mathbf{B}$ is the corresponding target data matrix, of dimension $k \times N$ (k is the dimension of the target data). The aim is to find the $k \times m$ linear mapping $\mathbf{M}$ that makes the approximation as good as possible.

Now, let us suppose small fluctuations in the target data, denoted as a matrix $\mathbf{E}$ of very small numbers (all entries are close to 0). The following regression is almost identical as Equation (1):

$$\mathbf{M}'\mathbf{A} \approx \mathbf{B} + \mathbf{E} \ . \tag{2}$$

And yet, often we find that the best solution for $\mathbf{M}$ and $\mathbf{M}'$ are very different from one another. The reason for this is that some dimensions of the measured data (the rows of the data matrix $\mathbf{A}$) could be highly correlated, such that there can be very different $\mathbf{M}$'s that fit $\mathbf{B}$ equally well.

Regularization theory [24] is a way of dealing with this kind of non-robustness. Given the example of Equation (1) and (2), we may ask that among all plausible (near optimized) solutions, which one is more likely to generalize to the 'unseen data' better. Typically, the principle of regularization follows the idea that the best fitting function (*i.e.* $\mathbf{M}$) should be the *simplest possible solution* that can still fits the data well.

In Fig. 2 we show a 1-D example. In the least-squares sense, the wiggly red curve is found to best fit the training data (black data points). Yet intuitively, this is not the correct fit, as the data points appear to follow a much simpler distribution. In contrast, the regularized fit (blue curve) seems to model the data better.

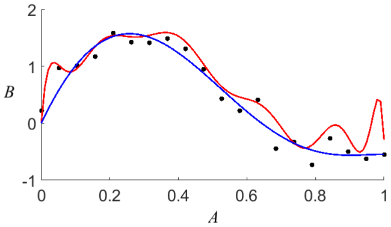

Fig. 2. Example of a least-squares fit (red curve) and a *regularized* least-squares fit (blue curve). The black data points are the training data. In the least-squares sense, the overall distance between the data points and the red wiggly fit is less than the blue smooth fit, but the latter looks 'more correct'.

Returning to our spectral reconstruction problem, taking *linear* regression as an example, the matrix $\mathbf{A}$ corresponds to a set of image RGBs ($m = 3$), and $\mathbf{B}$

refers to the spectra we are trying to recover (k is the number of spectral channels we have measured). The insight that we explore in this paper is that there is no reason why the fits for different spectral channels should be regularized altogether. Rather, we seek to consider the *per-channel* regression problem, where each spectral channel is recovered in turn, and correspondingly, each row of $\mathbf{M}$ is solved in turn. This simple modification allows us to carry out a *per-channel regularization* that ensures individual optimizations for all spectral channels in the spectral reconstruction problem.

Either for the conventional global regularization or for our per-channel approach, care must be taken not to overly tune the terms in $\mathbf{M}$ to the data at hand. This led us to develop a modified *cross validation* procedure. Our method separates the data on hand into three subsets, respectively for training, regularization and testing, which is a novel adjustment from the standard methodology [20] and is another contribution of this paper.

2 Background

2.1 Hyperspectral Images and RGB Simulation

In a hyperspectral image, spectra are measured *discretely* at some sampled wavelengths. Suppose the visible spectrum runs from 400 to 700 nanometers and the spectral sampling is every 10 nanometers, we get a 31-dimensional discrete representation of spectra, denoted as $\underline{\mathbf{r}} \in \mathbb{R}^{31}$.

Correspondingly, the spectral sensitivities of the R, G and B camera sensors can also be represented in discrete vector form (*i.e.* as 31-dimensional vectors), respectively denoted as $\underline{\mathbf{s}}_R$, $\underline{\mathbf{s}}_G$ and $\underline{\mathbf{s}}_B$. Then, as per the illustration in Fig. 1 we can write image formation as [26]:

$$\underline{\mathbf{x}} = \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{s}}_R^{\mathsf{T}} \\ \underline{\mathbf{s}}_G^{\mathsf{T}} \\ \underline{\mathbf{s}}_B^{\mathsf{T}} \end{pmatrix} \underline{\mathbf{r}} \ , \tag{3}$$

where $\underline{\mathbf{x}} = (R, G, B)^{\mathsf{T}}$ is the 3-value RGB camera response.

In the SR problem (the bottom of Fig. 1) we seek to recover hyperspectral images from the RGB images. Denote an SR algorithm as $\Psi : \ \mathbb{R}^3 \mapsto \mathbb{R}^{31}$,

$$\Psi(\underline{\mathbf{x}}) \approx \underline{\mathbf{r}} \ . \tag{4}$$

2.2 Regression-Based Spectral Reconstruction

The general regression-based formulation of the SR problem is written as

$$\Psi(\underline{\mathbf{x}}) = \mathbf{M}\varphi(\underline{\mathbf{x}}) \ , \tag{5}$$

where $\varphi(\cdot)$ is a *feature transformation* which maps each RGB to a corresponding p-term feature vector, and in turn it is mapped by the *regression matrix*

$\mathbf{M}$. Each of the various regression-based models [15, 11, 17, 18, 2] adopts a bespoke definition of $\varphi(\underline{\mathbf{x}})$. For details of the considered models, including Linear, Root-Polynomial and Adjusted Anchored Neighborhood Regression [15, 17, 2], see Appendix A.

Least-Squares Optimization The most common *least-squares* optimization seeks to minimize the *sum of squared errors* between the ground-truth training spectral data and the reconstruction from their corresponding RGBs: $\Psi(\underline{\mathbf{x}})$. Given the formulation of $\Psi(\underline{\mathbf{x}})$ in Equation (5), the least-squares optimization of $\mathbf{M}$ is formulated as:

$$\mathbf{M} = \arg\min_{\mathbf{M}} \left(\sum_{i=1}^{N} ||\underline{\mathbf{r}}_i - \mathbf{M}\varphi(\underline{\mathbf{x}}_i)||_2^2 \right) , \tag{6}$$

where N is the number of data points in the training set and i indexes an individual spectrum.

Collating all spectral training data in a data matrix $\mathbf{R} = (\underline{\mathbf{r}}_1, \underline{\mathbf{r}}_2, ..., \underline{\mathbf{r}}_N)$ and the corresponding feature vector matrix $\mathbf{\Phi} = (\varphi(\underline{\mathbf{x}}_1), \varphi(\underline{\mathbf{x}}_2), ..., \varphi(\underline{\mathbf{x}}_N))$, Equation (6) can be written as:

$$\mathbf{M} = \arg\min_{\mathbf{M}} \left(||\mathbf{R} - \mathbf{M}\mathbf{\Phi}||_F^2 \right) . \tag{7}$$

Here $|| \cdot ||_F^2$ is the squared *Frobenius norm*, which is exactly the sum-of-squares of all entries of the enclosed matrix.

Tikhonov Regularization In regression-based SR, the most common method to regularize a model is Tikhonov Regularization [15, 24], which hypothesizes that a more *natural* fit is obtained when the 'magnitude' (or the 'matrix norm') of $\mathbf{M}$ is bounded to some extent. Based on this assumption, the least-squares optimization in Equation (7) is extended to incorporate a regularization term:

$$\mathbf{M} = \arg\min_{\mathbf{M}} \left(||\mathbf{R} - \mathbf{M}\mathbf{\Phi}||_F^2 + \gamma||\mathbf{M}||_F^2 \right) . \tag{8}$$

Here, the $||\mathbf{M}||_F^2$ term (the regularization term, or *penalty* term) is controlled by a user-defined *regularization parameter* $\gamma \geq 0$, which is usually determined empirically [13, 21].

Equation (8) is solved in closed form [15]:

$$\mathbf{M} = \mathbf{R}\mathbf{\Phi}^{\mathsf{T}}(\mathbf{\Phi}\mathbf{\Phi}^{\mathsf{T}} + \gamma\mathbf{I})^{-1} , \tag{9}$$

where $\mathbf{I}$ is the $p \times p$ identity matrix (recall that p is the dimension of the feature vectors $\varphi(\underline{\mathbf{x}})$).

3 Proposed Method

In spectral reconstruction, we wish to recover spectral measurements in the range from 400 to 700 nanometers (the visible spectrum). Suppose we know the intensity of light entering the camera at 400 nanometers. Given this knowledge if we wished to predict the value of the spectrum at 410 nanometers, it makes sense to assume a similar value as the one at 400 nanometers. Indeed, the fact that the intensity values at close-by wavelengths are similar is why we can represent the continuous visible spectrum at discrete wavelengths. Conversely, one could not use the knowledge of light at 400 nanometers to predict the spectral value at, say, 700 nanometers.

And yet, in the literature when we regularize the regression-based SR models, we are - in some sense - assuming that all wavelengths are related. Our new per-channel reformulation of Tikhonov regularization for spectral reconstruction effectively allows the recovery of the spectral values at distant wavelengths to be considered more independently from one another.

3.1 Per-Channel Regularization

Let us split the regression matrix $\mathbf{M}$ by row: $\mathbf{M} = (\underline{\mathbf{m}}_1, \underline{\mathbf{m}}_2, ..., \underline{\mathbf{m}}_{31})^\intercal$, such that the general form of regression-based SR formulated in Equation (5) becomes

$$\Psi(\underline{\mathbf{x}}) = \mathbf{M}\varphi(\underline{\mathbf{x}}) = \begin{pmatrix} \underline{\mathbf{m}}_1^\intercal \\ \underline{\mathbf{m}}_2^\intercal \\ \vdots \\ \underline{\mathbf{m}}_{31}^\intercal \end{pmatrix} \varphi(\underline{\mathbf{x}}) = \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_{31} \end{pmatrix} , \qquad (10)$$

where $(\hat{r}_1, \hat{r}_2, ..., \hat{r}_{31})^\intercal = \underline{\hat{\mathbf{r}}}$ is the reconstructed spectrum. For an arbitrary k^{th} spectral channel, the estimated intensity value $\hat{r}_k$ is given by $\hat{r}_k = \underline{\mathbf{m}}_k^\intercal \varphi(\underline{\mathbf{x}})$.

Note that as we represent the regression model by channel, we do **not** alter the original model. This says that the regression-based spectral reconstruction has always been in such a way that the reconstruction for each spectral channel depends *exclusively* on the corresponding row of $\mathbf{M}$.

Given this fact, we might expect that each row of $\mathbf{M}$ would be optimized independently. However, this was not the case for the conventional regularized least-squares solution in Equation (9). Indeed, we see in Equation (8) the regularization term is only controlled by one single regularization parameter γ and the fits for all spectral channels (all rows of $\mathbf{M}$) are regularized by the same γ. Regardless of how we optimize this γ parameter, this setting clearly contradicts to the inherent independence between the rows of $\mathbf{M}$.

Let us split the spectral reconstruction problem into 31 independent problems, where the function $\Psi_k : \mathbb{R}^3 \mapsto \mathbb{R}$ reconstructs the k^{th}-channel values of the reconstructed spectra by the k^{th} row of $\mathbf{M}$:

$$\Psi_k(\underline{\mathbf{x}}) = \underline{\mathbf{m}}_k^\intercal \varphi(\underline{\mathbf{x}}) \ . \qquad (11)$$

Then, we are to determine $\underline{\mathbf{m}}_k^\intercal$, as the least-squares fit for the k^{th} channel data. Recall the training spectral data matrix $\mathbf{R}$ whose columns are individual training spectra, now we split $\mathbf{R}$ by spectral channel instead:

$$\mathbf{R} = (\underline{\mathbf{r}}_1, \underline{\mathbf{r}}_2, ..., \underline{\mathbf{r}}_N) = \begin{pmatrix} \underline{\boldsymbol{\rho}}_1^\intercal \\ \underline{\boldsymbol{\rho}}_2^\intercal \\ \vdots \\ \underline{\boldsymbol{\rho}}_{31}^\intercal \end{pmatrix} \ , \tag{12}$$

where $\underline{\boldsymbol{\rho}}_k^\intercal$ includes the k^{th} channel values of all training spectral data. $\underline{\mathbf{m}}_k^\intercal$ is then optimized following the regularized least-squares optimization:

$$\underline{\mathbf{m}}_k^\intercal = \arg\min_{\underline{\mathbf{m}}_k^\intercal} \left(||\underline{\boldsymbol{\rho}}_k^\intercal - \underline{\mathbf{m}}_k^\intercal \mathbf{\Phi}||_2^2 + \gamma_k ||\underline{\mathbf{m}}_k||_2^2 \right) \ , \tag{13}$$

and solved in closed form:

$$\underline{\mathbf{m}}_k^\intercal = \underline{\boldsymbol{\rho}}_k^\intercal \mathbf{\Phi}^\intercal (\mathbf{\Phi}\mathbf{\Phi}^\intercal + \gamma_k \mathbf{I})^{-1} \ . \tag{14}$$

Here γ_k represents the channel-wise regularization parameter that only controls the regularization for the k^{th} channel.

Clearly, our per-channel regularization scheme (Equation (14)) solves the regression matrix $\mathbf{M}$ row-by-row, such that each row is ready to be regularized independently. The remaining question is then how we are going to optimize these regularization parameters.

3.2 Modified Cross Validation

Perforce, regularization parameters (conventionally the single γ and our per-channel γ_k's) are determined empirically. In the literature, a *grid-search* approach is adopted, where different parameters are tried to regularize the model. These 'intermediate models' are then used to recover spectra from a set of unseen RGB images, and the model that minimizes the evaluation criteria is selected. For example, see [11, 17, 2].

As usual we would like to train, regularize and test a model using the images from the same database, we must partition the database into several subsets for these different usages. All (to our knowledge) deep-learning models simply separate the image database into 3 subsets randomly (respectively for training, *validation* and testing, in the parlance of deep learning), see [22, 6]. However, this setting can potentially create so-called 'unfair' separations, such that if the database is separated differently, the results may vary.

A better practice is using a *cross-validation* process. In this paper we develop our own cross validation scheme, which is modified from the conventional K-fold cross validation [20]. This is because the conventional K-fold only seeks to separate a dataset into training and testing data, and here we need an additional partition as the regularization data.

In Fig. 3 we show the comparison between the conventional 4-fold cross validation (left) and our method (right). For both methods the same experiment is conducted 4 times. In each trial, the conventional method selects 3 out of 4 portions of data for training (marked in blue) and the remaining part is for testing (marked in orange). In our method, however, only 2 out of 4 portions of data are for training, which allows 1 portion of data (marked in green) used for regularization, that is to determine the γ and γ_k parameters. Subject to these terms we solve for the best regression model for the training (blue) data. The performance statistics are calculated based on the recovery errors on the testing (orange) data and averaged over the 4 trials.

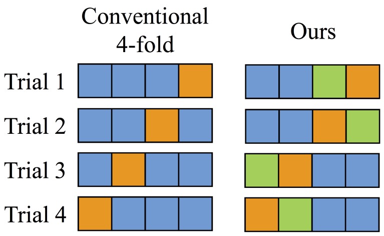

Fig. 3. The conventional 4-fold cross validation (left) and the modified scheme used in this paper (right). Each colored patch represents equal amount of randomly allocated data. The blue, orange and green patches represent the data for training, testing and regularization, respectively.

Notice for our cross validation method there actually exists more possible permutations than the presented 4-trial setting. To be exact, there should be 12 different permutations. We remark that according to our empirical study, experimenting with more trials (than the presented setting) does not make significant difference in the performance statistics.

4 Experiments

4.1 Considered Models

In this paper we consider 3 regression-based models:

- Linear Regression (LR) [15]
- Root-Polynomial Regression (RPR) [17]
- Adjusted Anchored Neighborhood Regression (A+) [2] .

For all the above models we adopt both the *original* regularization (as reported in their citations and as per Equation (9)) and our per-channel regularization (as per Equation (14)).

4.2 Database

We use the ICVL hyperspectral image database [3] (Fig. 4), which provides 201 hyperspectral images of spatial dimension 1392 × 1300 and with 31 spectral channels. The spectral channels represent narrow-band intensity measurements, respectively at every 10 nanometers (nm) between 400 and 700 nm.

The corresponding RGB images are simulated following the linear RGB simulation setting (Equation (3)), and the CIE 1964 color matching functions [10] are used as the spectral sensitivities.

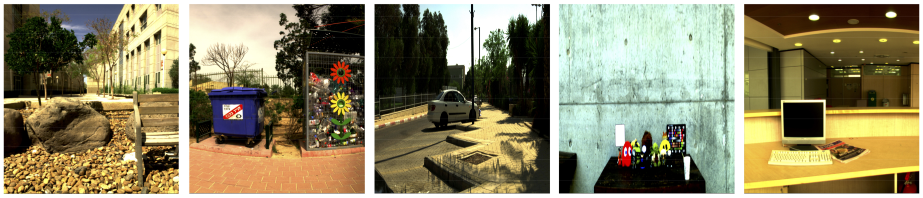

Fig. 4. Example scenes from the ICVL hyperspectral database [3]. Note that the shown RGB images are rendered only for display (not the ground-truth RGB images that used in the experiments).

4.3 Evaluation Criteria

The selected evaluation metric is Relative Absolute Error (RAE) [4, 5], which is defined per channel as:

$$RAE(r_k, \hat{r}_k) = \left| \frac{r_k - \hat{r}_k}{r_k} \right| , \tag{15}$$

where r_k and $\hat{r}_k$ are respectively the k^{th}-channel values of the ground-truth and reconstructed spectra. Effectively, this metric measures the *percentage* absolute error. RAE is the most common performance measure used in recent research, and the rationale of using this metric can be found in [4].

5 Results and Discussion

In Table 1, we present the per-*channel* error statistics of LR (left table), RPR (middle table) and A+ (right table) under the original settings - where a single penalty term is used in the regularization [15, 17, 2] - and our per-channel regularization method. The spectral channels are represented by the wavelengths λ (nm). We also calculate the percentage 'improvements' as:

$$\text{Improve (\%)} = 100 \times \frac{RAE_{original} - RAE_{ours}}{RAE_{original}} , \tag{16}$$

Table 1. Spectral reconstruction results at each spectral channel for LR (left), RPR (middle) and A+ (right). Significant improvements (> 10%) are marked in bold and with underlines. For each model, the Mean RAE error over all spectral channels are given in the bottom of each table and marked in gray.

LR			
λ	Heikkinen *et al.* [15]	**Ours**	**Improve**
nm	RAE ($\times 10^{-2}$)		%
400	26.96	26.89	0.3%
410	17.63	17.57	0.4%
420	12.13	12.08	0.4%
430	10.10	10.05	0.4%
440	4.05	4.02	0.8%
450	2.14	2.07	3.4%
460	5.07	5.06	0.1%
470	7.06	7.04	0.3%
480	8.83	8.79	0.4%
490	8.16	8.12	0.6%
500	7.47	7.42	0.7%
510	5.49	5.43	1.1%
520	1.71	1.68	1.6%
530	1.72	1.71	0.9%
540	3.30	2.96	**10.4%**
550	4.04	3.58	**11.4%**
560	4.07	3.56	**12.4%**
570	3.18	2.64	**16.9%**
580	1.78	1.47	**17.4%**
590	1.51	1.48	1.6%
600	1.10	1.10	0.1%
610	2.82	2.55	9.7%
620	4.05	3.56	**12.1%**
630	3.99	3.52	**11.7%**
640	5.08	4.51	**11.2%**
650	4.87	4.60	5.5%
660	5.34	5.02	6.0%
670	7.04	6.58	6.4%
680	6.79	6.56	3.3%
690	5.72	5.62	1.6%
700	10.39	10.31	0.8%
All	6.24	6.05	3.1%

RPR			
λ	Lin *et al.* [17]	**Ours**	**Improve**
nm	RAE ($\times 10^{-2}$)		%
400	21.15	19.51	7.8%
410	13.36	12.27	8.2%
420	9.31	7.83	**15.9%**
430	7.51	6.40	**14.7%**
440	2.91	2.50	**14.0%**
450	1.87	1.62	**13.3%**
460	3.50	3.12	**10.9%**
470	4.94	4.18	**15.4%**
480	6.05	5.32	**12.1%**
490	5.31	4.88	8.1%
500	5.05	4.68	7.3%
510	3.73	3.59	3.6%
520	1.73	1.58	8.8%
530	1.27	1.22	4.0%
540	2.48	2.28	8.1%
550	3.03	2.75	9.3%
560	2.72	2.63	3.4%
570	2.30	1.93	**15.8%**
580	1.23	1.14	6.9%
590	1.11	1.02	8.0%
600	1.02	0.96	5.9%
610	1.93	1.82	6.2%
620	2.82	2.35	**16.9%**
630	2.76	2.50	9.4%
640	3.30	3.17	3.9%
650	3.76	3.28	**12.8%**
660	4.05	3.93	3.1%
670	5.44	5.17	5.0%
680	5.55	5.24	5.5%
690	5.51	5.45	1.1%
700	8.96	8.88	0.9%
All	4.70	4.30	8.6%

A+			
λ	Aeschbacher *et al.* [2]	**Ours**	**Improve**
nm	RAE ($\times 10^{-2}$)		%
400	16.01	16.08	-0.5%
410	10.41	10.43	-0.2%
420	7.10	7.11	-0.1%
430	5.68	5.68	0.0%
440	2.28	2.23	2.3%
450	1.54	1.28	**16.7%**
460	2.83	2.47	**12.6%**
470	3.80	3.40	**10.5%**
480	4.70	4.32	8.1%
490	4.41	4.16	5.7%
500	4.11	3.94	4.1%
510	3.12	3.01	3.4%
520	1.31	1.27	2.6%
530	1.16	1.15	1.3%
540	1.98	1.97	0.6%
550	2.38	2.36	0.8%
560	2.33	2.31	0.7%
570	1.82	1.80	0.9%
580	1.18	1.16	1.8%
590	1.06	1.05	1.2%
600	0.88	0.88	0.6%
610	1.62	1.55	4.2%
620	2.17	2.07	4.8%
630	2.26	2.15	4.8%
640	3.04	2.80	7.7%
650	3.31	3.12	5.9%
660	3.86	3.68	4.6%
670	5.23	4.81	8.0%
680	5.16	4.79	7.3%
690	4.66	4.65	0.3%
700	8.01	7.97	0.6%
All	3.85	3.73	3.2%

which is presented in the rightmost column of each table. In the bottom of each table, the Mean RAE results (averaging over all spectral channels) are shown.

First, we see that for all considered models, our method improves the RAE in multiple channels by over 10% (marked in bold and with underlines), with maximal improvements around 16-17%.

Secondly, in terms of Mean RAE performance, our method improves the RPR model the most, by an 8.6% increment, compared to 3.2% for A+ and 3.1% for LR. Significantly, the A+ model is the leading sparse coding model, which is shown able to perform equally-well as some deep-learning solutions [2]. By improving the A+ model, we effectively bring forward the shallow-learned baseline. Moreover, our method reduces the gap between RPR and A+. Relative to A+, RPR model is much simpler (with significantly less model parameters) [17], which ensures more effective model re-training and shorter runtime.

Lastly, for the A+ model, it seems curious that the per-channel performances in the first three channels (400, 410 and 420 nm) degrade by minute differences. Indeed, this means the regularization parameters we chose for these channels are not actually optimized for the test-set data. We remark that this is most likely originated from the unequal separation of data subsets in cross validation, such that the best regularization parameter for the regularization-set data does not correspond to the best for the test-set data. We are investigating how to remedy this issue.

For one example image in the ICVL database [3], we visualize the spectral reconstruction errors as the Mean RAE error maps in Fig. 5. It is clear that for all models our method improves the Mean RAE in various parts of the image. For example, the tree stem for LR and RPR, and the grassy ground for A+.

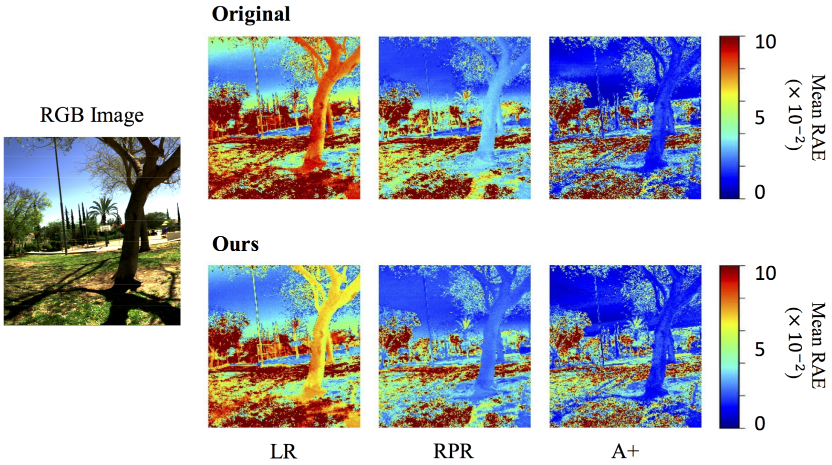

Fig. 5. Visualizing the spectral reconstruction outcome of the considered models under the original setting (top row) and our new regularization method (bottom row). The RGB image (left) is rendered for illustration purpose, which is not the RGBs used for spectral reconstruction.

6 Conclusion

In the spectral reconstruction (SR) problem, hyperspectral images are reconstructed from RGB images. Many approaches are based on *least-squares regression*, where the fitting function is modelled by a simple linear transformation, and a *Tikhonov regularization* process is applied to improve model generalizability. Conventionally, the fits for all spectral channels are jointly regularized. We demonstrate that the fit for each spectral channel can be formulated independently, such that the fit for each channel is regularized (therefore optimized) independently. We also provide a novel modification of K-fold cross validation so that the models can be fairly trained, regularized and tested with a single image database. Compared to the original models, our per-channel regularization method improves the accuracy of recovery for individual spectral channels by up to 17% increments, and by 3-9% in mean improvements over all spectral channels.

References

1. Abrardo, A., Alparone, L., Cappellini, I., Prosperi, A.: Color constancy from multispectral images. In: Proceedings of the International Conference on Image Processing. vol. 3, pp. 570–574. IEEE (1999)
2. 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. pp. 471–479. IEEE (2017)
3. Arad, B., Ben-Shahar, O.: Sparse recovery of hyperspectral signal from natural RGB images. In: Proceedings of the European Conference on Computer Vision. pp. 19–34. Springer (2016)
4. Arad, B., Ben-Shahar, O., Timofte, R., et al.: NTIRE 2018 challenge on spectral reconstruction from RGB images. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops. pp. 929–938. IEEE (2018)
5. Arad, B., Timofte, R., Ben-Shahar, O., Lin, Y., Finlayson, G.D., et al.: NTIRE 2020 challenge on spectral reconstruction from an RGB image. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops. IEEE (2020)
6. Arun, P.V., Buddhiraju, K.M., Porwal, A., Chanussot, J.: CNN based spectral super-resolution of remote sensing images. Signal Processing **169**, 107394 (2020)
7. Bashkatov, A.N., Genina, E.A., Kochubey, V.I., Tuchin, V.V.: Optical properties of the subcutaneous adipose tissue in the spectral range 400–2500 nm. Optics and Spectroscopy **99**(5), 836–842 (2005)
8. Chen, Y., Zhao, X., Jia, X.: Spectral–spatial classification of hyperspectral data based on deep belief network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **8**(6), 2381–2392 (2015)
9. Cheung, V., Westland, S., Li, C., Hardeberg, J.Y., Connah, D.R.: Characterization of trichromatic color cameras by using a new multispectral imaging technique. Journal of the Optical Society of America A **22**(7), 1231–1240 (2005)
10. Commission Internationale de l'Eclairage: CIE proceedings 1964 Vienna session, committee report E-1.4. 1 (1964)
11. Connah, D.R., Hardeberg, J.Y.: Spectral recovery using polynomial models. In: Color Imaging X: Processing, Hardcopy, and Applications. vol. 5667, pp. 65–75. International Society for Optics and Photonics (2005)
12. Finlayson, G.D., Morovic, P.: Metamer sets. Journal of the Optical Society of America A **22**(5), 810–819 (2005)

13. Galatsanos, N.P., Katsaggelos, A.K.: Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. IEEE Transactions on Image Processing **1**(3), 322–336 (1992)
14. Ghamisi, P., Dalla Mura, M., Benediktsson, J.A.: A survey on spectral–spatial classification techniques based on attribute profiles. IEEE Transactions on Geoscience and Remote Sensing **53**(5), 2335–2353 (2014)
15. 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. Journal of the Optical Society of America A **25**(10), 2444–2458 (2008)
16. Lam, A., Sato, I.: Spectral modeling and relighting of reflective-fluorescent scenes. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. pp. 1452–1459. IEEE (2013)
17. Lin, Y., Finlayson, G.D.: Exposure invariance in spectral reconstruction from RGB images. In: Proceedings of the Color and Imaging Conference. vol. 2019, pp. 284–289. Society for Imaging Science and Technology (2019)
18. Nguyen, R.M.H., Prasad, D.K., Brown, M.S.: Training-based spectral reconstruction from a single RGB image. In: Proceedings of the European Conference on Computer Vision. pp. 186–201. Springer (2014)
19. Pan, Z., Healey, G., Prasad, M., Tromberg, B.: Face recognition in hyperspectral images. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(12), 1552–1560 (2003)
20. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-Validation. In: Encyclopedia of Database Systems, pp. 532–538. Springer US, Boston, MA (2009)
21. Regińska, T.: A regularization parameter in discrete ill-posed problems. SIAM Journal on Scientific Computing **17**(3), 740–749 (1996)
22. 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. pp. 939–947. IEEE (2018)
23. Tao, C., Pan, H., Li, Y., Zou, Z.: Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. IEEE Geoscience and Remote Sensing Letters **12**(12), 2438–2442 (2015)
24. Tikhonov, A.N., Goncharsky, A.V., Stepanov, V.V., Yagola, A.G.: Numerical Methods for the Solution of Ill-posed Problems, vol. 328. Springer Science & Business Media (2013)
25. Veganzones, M.A., Tochon, G., Dalla-Mura, M., Plaza, A.J., Chanussot, J.: Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation. IEEE Transactions on Image Processing **23**(8), 3574–3589 (2014)
26. Wandell, B.A.: The synthesis and analysis of color images. IEEE Transactions on Pattern Analysis and Machine Intelligence (1), 2–13 (1987)
27. Xu, P., Xu, H., Diao, C., Ye, Z.: Self-training-based spectral image reconstruction for art paintings with multispectral imaging. Applied Optics **56**(30), 8461–8470 (2017)
28. Zhang, Y., Mou, X., Wang, G., Yu, H.: Tensor-based dictionary learning for spectral CT reconstruction. IEEE Transactions on Medical Imaging **36**(1), 142–154 (2016)
29. Zhang, Y., Xi, Y., Yang, Q., Cong, W., Zhou, J., Wang, G.: Spectral CT reconstruction with image sparsity and spectral mean. IEEE Transactions on Computational Imaging **2**(4), 510–523 (2016)

A Appendix: Regression Models

A.1 Linear Regression

Linear regression (LR) [15] assumes a linear map from RGB to spectra. The spectral estimate is written as

$$\Psi(\underline{\mathbf{x}}) = \mathbf{M}\underline{\mathbf{x}} \ , \tag{17}$$

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

A.2 Root-Polynomial Regression

As a simple non-linear extension from LR, in Root-Polynomial Regression (RPR) [17] we expand a series of root-polynomial terms from each RGBs. Denote $\varphi^\alpha : \mathbb{R}^3 \mapsto \mathbb{R}^p$ as the α-order root-polynomial transformation, the example transformations for the $2^{nd}, 3^{rd}$ and 4^{th} order RPR are:

Order	**Root-Polynomials**
$\varphi^2(\underline{\mathbf{x}})$	$R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{BR}$
$\varphi^3(\underline{\mathbf{x}})$	$R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{BR},$ $\sqrt[3]{RG^2}, \sqrt[3]{GB^2}, \sqrt[3]{BR^2}, \sqrt[3]{R^2G}, \sqrt[3]{G^2B}, \sqrt[3]{B^2R}, \sqrt[3]{RGB}$
$\varphi^4(\underline{\mathbf{x}})$	$R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{BR},$ $\sqrt[3]{RG^2}, \sqrt[3]{GB^2}, \sqrt[3]{BR^2}, \sqrt[3]{R^2G}, \sqrt[3]{G^2B}, \sqrt[3]{B^2R}, \sqrt[3]{RGB},$ $\sqrt[4]{R^3G}, \sqrt[4]{R^3B}, \sqrt[4]{G^3R}, \sqrt[4]{G^3B}, \sqrt[4]{B^3R}, \sqrt[4]{B^3G},$ $\sqrt[4]{R^2GB}, \sqrt[4]{G^2RB}, \sqrt[4]{B^2RG}$

The spectral reconstruction then seeks to linearly map these root-polynomial vectors to spectra:

$$\Psi(\underline{\mathbf{x}}) = \mathbf{M}\varphi^\alpha(\underline{\mathbf{x}}) \ . \tag{18}$$

In this paper we set $\alpha = 6$, which is the 6^{th}-order RPR.

A.3 Adjusted Anchored Neighborhood Regression (A+ Sparse Coding)

The leading sparse coding method 'A+' [2] assumes linear maps from RGBs to spectra (effectively, operates LR in every neighborhood). Denote $\Psi^i(\underline{\mathbf{x}})$ as the spectral reconstruction mapping for the data in the i^{th} neighborhood. On input of an RGB $\underline{\mathbf{x}}$, the reconstruction is written as:

$$neighborhood(\underline{\mathbf{x}}) = i \ \Rightarrow \ \Psi^i(\underline{\mathbf{x}}) = \mathbf{M}^i\underline{\mathbf{x}} \ . \tag{19}$$

See [2] for more details about the model.