A Practical Study on Recovering Spectra from RGB Images

Yi-Tun Lin

A thesis presented for the degree of
Doctor of Philosophy

University of East Anglia, United Kingdom
School of Computing Sciences
October, 2023

A Practical Study on Recovering Spectra from RGB Images

Yi-Tun Lin

2022

Abstract

RGB cameras make three measurements of the light entering the camera, whereas hyperspectral imaging devices, per pixel, record the spectrum of the light. Spectral images have been shown to be more useful than RGB images in solving problems in many industrial application areas, including remote sensing and medical imaging. Spectral Reconstruction (SR) refers to a computational algorithm that recovers spectra from the RGB camera responses. This "make-the-RGBs-more-informative" process is most commonly implemented by machine learning (ML) algorithms, given matching RGB and hyperspectral data for training. Two mainstream ML approaches used in SR are regression and Deep Neural Network (DNN). While the former often has simple closed-form formulations for a pixel-based mapping, the latter approach is much more complicated: millions of parameters are used to map large image patches, in the hope that the network could utilise the spatial context in which each RGB is seen to further improve SR. It is generally accepted that regressions have long since been superseded by DNN methods. Nevertheless, few studies have actually been dedicated to comparing the two approaches.

There are three main goals of this thesis. First, we benchmark regression- and DNN-based SR algorithms on the same hyperspectral image dataset. Here we pay close attention to the

role that the spectral sensitivities of a camera play and also SR performance on unseen data. Second, we seek to improve regression-based algorithms and, in effect, attempt to close their gap in performance compared to DNN counterparts. Lastly, we investigate the practical issues faced by all SR algorithms. We consider SR performance as exposure changes and SR performance in a “closed-loop” imaging framework (i.e., do the spectra that an SR algorithm recovers integrate to the same input RGBs?).

Our baseline benchmarking experiments indicate that the best DNN method only delivers a 12% accuracy improvement compared to the best-performing regression. Moreover, a regression method trained for one camera might actually outperform a DNN trained on another camera. Additionally, we find that the DNN’s worst-case performance (for unseen and unexpected scenes) is no better than the simplest regression method. Concomitantly, this encourages us to see if we could improve the average performance of regression methods.

We propose three new improvements for regression methods. First, we reformulate the regressions so that they minimise a loss metric that is more similar to the one used to rank and train the leading DNN methods. Secondly, we revisit the regularisation step of the regression implementation. Regularisation is a technique for making the outputs of regressions more stable for unseen input and is usually governed by a single regularisation parameter. Here, we adopt as many regularisations as there are channels in a hyperspectral image, and this results in significant performance improvement. Lastly, we propose a new sparse regression framework. In sparse regression, we code RGBs in terms of the neighbourhood in the RGB space (via a clustering argument). We argue that this clustering is better performed in the spectral domain (where input RGBs are first regressed to some primary estimation of spectra). Combined, upgraded formulation and improved clustering, we develop a regression-based method found to work as well as the top DNN methods.

As important as spectral accuracy is, trained SR algorithms need to work in practice, e.g., where objects and scenes can be viewed in varying exposure conditions. Unfortunately, we

find that leading methods, such as non-linear regressions and DNNs, do not work well when exposure changes. Consequently, we propose new training frameworks which ensure the DNNs and regressions continue to work well under changing exposures.

Finally, we investigate the following problem: we find that both regression- and DNN-based SR algorithms recover spectra that—when integrated with the camera's spectral sensitivities—do not induce the same RGBs as the input to the algorithm. This means that the spectra that are recovered cannot (ever) be the correct spectra. Given this finding, we seek ways of adding physical plausibility (spectra should integrate to predict the input RGBs) to the SR algorithms. One of our proposed solutions is effectively a simple post-processing step which, provably, always improves the RMS (i.e., root-mean-square) performance of any SR algorithm.

Access Condition and Agreement

Each deposit in UEA Digital Repository is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the Data Collections is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form. You must obtain permission from the copyright holder, usually the author, for any other use. Exceptions only apply where a deposit may be explicitly provided under a stated licence, such as a Creative Commons licence or Open Government licence.

Electronic or print copies may not be offered, whether for sale or otherwise to anyone, unless explicitly stated under a Creative Commons or Open Government license. Unauthorised reproduction, editing or reformatting for resale purposes is explicitly prohibited (except where approved by the copyright holder themselves) and UEA reserves the right to take immediate 'take down' action on behalf of the copyright and/or rights holder if this Access condition of the UEA Digital Repository is breached. Any material in this database has been supplied on the understanding that it is copyright material and that no quotation from the material may be published without proper acknowledgement.

A Practical Study on Recovering Spectra from RGB Images

Yi-Tun Lin

© This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that use of any information derived there-from must be in accordance with current UK Copyright Law. In addition, any quotation or extract must include full attribution.

Contents

List of Tables

List of Figures

Acknowledgements

My biggest thanks go to my Ph.D. advisor, Prof. Graham D. Finlayson. His vision, knowledge and experience continue to inspire me throughout this journey. I am especially grateful for his dedication in guiding me to become a mature and independent researcher. I also want to thank Dr. Michal Mackiewicz for his constant assistance and supervision, and Apple Inc. for funding my Ph.D. tuition. Thanks also go to my two external examiners, Prof. Ohad Ben-Shahar and Dr. William Smith, for the in-depth discussions during the Viva and helpful suggestions for my thesis.

Many thanks to my colleagues in the Colour and Imaging Lab: Fufu Fang, Ghalia Hemrit, Yuteng Zhu, Ellen Bowler, Jake McVey, Chloe Game, Artjom Gorpincenko and Brandon Hobley, for their great companies and moral support. Special thanks to Dr. Javier Vazquez-Corral and Geoffrey French for their professional guidance and friendship.

I would also like to thank for the constant support and unconditional love I receive from my family. My friends, you also have my gratitude: Krista, Joe, Kev, Karissma, Ingrid, Benny and Eddie. Finally, my heartfelt thanks go to my beloved partner, Alexandra, for being the greatest company I could ask for.

Chapter 1

Introduction

An RGB camera records the light radiances coming from the scene with 3 types of colour sensors, which, effectively, records the weighted-sums of the spectral intensities in broadly 3 spectral regions—Red, Green and Blue [93]. This almost ubiquitous imaging practice sacrifices a significant portion of spectral information for faster, light weight and more affordable image captures. On the other hand, there are hyperspectral cameras that measure the light radiances with high "spectral" resolution. With more detailed radiance's spectral shape recorded, hyperspectral cameras are more able to tell the light coming from different objects apart—a.k.a. the concept of "spectral signature" [44]—and these devices are found to be more useful (compared with RGB cameras) for many industrial applications including, remote sensing [90; 97], medical imaging [26; 56], anomaly detection [54; 102], artwork conservation [100; 39; 67], device characterisation [23] and food processing [21; 36]. However, the high price tag of hyperspectral imaging technologies, as well as the trade-offs among spatial, spectral and temporal resolutions limit their usefulness.

To bridge the gap between RGB capture (simple, high spatial resolution and cheap) and hyperspectral imaging, there are various techniques that substitute physical per-pixel hypespectral capture with ingenious computational algorithms. Examples include hyperspectral image super-resolution [27; 43] (which enhances the spatial resolution of low-resolution hyperspectral images), compressive sensing techniques [34; 104] (jointly record spectral and spatial information in a 2-D plane and "decompress" using specialised algorithms), and Spectral Reconstruction (SR). SR is the focus of this thesis.

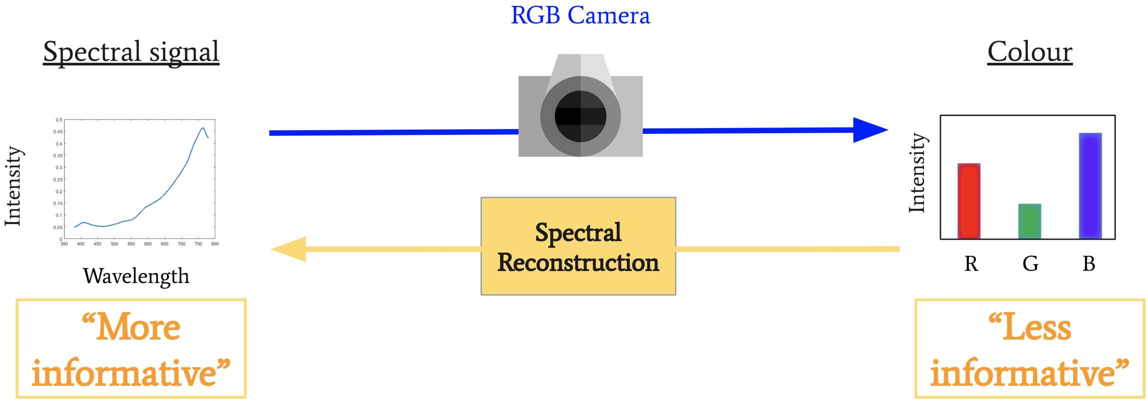


Figure 1.1: Illustration of colour image formation and spectral reconstruction (SR).

In SR, hyperspectral measurements are recovered from spectral images with lower number of spectral channels, typically from RGB images. As shown in Figure 1.1, an RGB camera only records three intensity values at each pixel (right side of the figure), whereas the spectral signal (left side), or the hyperspectral measurement of the pixel, is a *continuous* (or practically, finely sampled) function of wavelength.

Solving the SR-from-RGB problem has merit. First, in applications such as digital art archiving [100] and device spectral characterisation [23], acquiring the spectral representation of objects is often immediately the goal. Then, in other applications, e.g., computer vision, where we might want to use spectral images only as an intermediate representation for further objectives, given RGB images, it may be better to solve the problems end-to-end. Indeed, if hyperspectral images are *estimated* from the RGBs, the recovered spectra will contain no more information than the RGBs (as opposed to Figure 1.1 where the spectra were *recorded*). Still, with a controlled intermediate representation, it is likely that less data and less complex learning model will suffice for the tasks [61]. Another advantage of studying the SR problem is that, with a hyperspectral camera, the ground-truths for SR learning are immediately labelled, i.e., the spectral ground-truths captured by a spectral camera do not need further manual labelling.

In most contemporary research, SR algorithms attempt to recover the raw radiance spectra captured by a hyperspectral device within the *visible* range—in which the RGB camera sensors are responsive. Of course, if there is a single prevailing light illuminating a scene, and this illuminant's spectrum is known, then the *reflectance* spectra can be derived from the radiances by "dividing out" the known illuminant spectrum [93] and recovered using an SR algorithm. Though many earlier SR researches are based on reflectance recovery, following the recent trend, this thesis will focus on examining the RGB-to-radiance SR recovery, and the term "spectrum" will almost always refer to the radiance spectrum in this thesis.

Evidently, describing a pixel by its full spectral function—either radiance or reflectance—is more informative than by its colour. Yet, this also means that solving SR, the RGB-to-spectrum recovery, is an under-constrained and ill-posed problem [87]. In the prior art, SR algorithms are mostly formulated either as a regression problem [42; 25; 62; 1] or in a Deep Neural Network (DNN) framework [7; 8; 9]. Although other approaches exist, including Bayesian inference [60; 14] and iterative methods [13; 107], they are less efficient, with more complex implementation, and work no better than regression and poorer compared with state-of-the-art DNN algorithms.

In regression-based SR, a one-to-one map from RGBs (or simple non-linear features derived from the RGBs) to spectra is formulated using a linear transformation matrix, whose least-squares optimum can be solved in closed form. This ensures very fast training and inference. The variation of regressions used in SR literature includes the most primitive linear regression [42], the non-linear polynomial regression [25], the clustering-based radial-basis-function network [62] and A+ sparse coding method [1]. However, the one-to-one nature of the map neglects the fact that multiple (in fact *infinite*) plausible spectra could be the answer given an RGB input [60]. This phenomenon is so-called *metamerism* [29; 99]. In contrast, the recent DNN implementations are based on much stronger feature extraction and mapping architectures to provide much richer SR inferences. Additionally, DNN methods usually take large image patches as inputs, which provides the possibility for the algorithm to learn high-level descriptions of the pixels that could potentially overcome the metameric problem—i.e., if the same colour is viewed as part of two different objects, DNN can potentially distinguish the two cases and map to different spectral estimates.

Nevertheless, we observe that it is unclear in the literature how large the gap between regression-based methods and the leading DNN methods truly are. Indeed, most regression methods are proposed over 10 years ago and benchmarked on small dataset of discrete hyperspectral measurements, whereas the recent DNN methods are evaluated with the much larger hyperspectral "image" datasets—e.g., ICVL [5], NUS [62] and CAVE [101]—and only focus on comparing their methods with other DNN-based methods. Therefore, the first main focus of this thesis is to re-evaluate the regression-based methods on the ICVL hyperspectral image dataset, where a fair comparison with the *leading* DNN methods is possible. Here, I put *leading* in italic because, admittedly, the trend of proposing new DNN methods to solve SR does not stop even now as I am writing this thesis. Nevertheless, it is undoubtedly that whichever DNN methods we are comparing (the DNN methods that are the best at some point during the thesis development) are much more complicated than any of the regression methods involved.

As we redo the evaluation of regressions and leading DNN methods on the same evaluation protocol, we show that the current gap between the two approaches is around 12%. Further, we show that the DNN performance on a given camera $\mathcal{A}$ can be worse than that delivered by regression using another camera $\mathcal{B}$. In other words, it is possible to reach DNN's performance using a regression by switching the camera used in SR. Next, we develop a worst-case imaging condition to evaluate SR, where the image content and the relative abundance among spectra is removed. We show that in this generalisability testing, the leading DNNs do not outperform even the most primitive *linear regression* [42]. These results lay out an important empirical evidence that the difference between the old regression methods and the currently leading DNN methods may not be as large as we may think.

Encouraged by this outcome, we further investigate how we could improve the regression methods used in SR research, so as to narrow down their performance gap with the leading DNN methods. In the first attempt, we seek to reformulate regression so that it does not minimise the usual least-squares loss function, but one that is more similar to the percentage-error-type metric the DNNs are trained and benchmarked against [7; 8]. The methods we develop from this change-the-metric insights can be used, in general, for any regression scheme.

Our second contribution to regression theory is even more important and surprising. We re-examine the sparse regression SR approach [5; 1]. In its simplest guise, sparse regression first finds the *neighbourhood* in which an RGB resides. This is typically defined as the overlapping surrounding areas of a small number of control-point RGBs. Correspondingly, there are spectra associated with these RGBs. By solving how a given query RGB can be written as a linear combination of nearby control-point RGBs, we apply the same linear combination to the corresponding control-point spectra to estimate the spectrum. Then, our insight is to estimate how the linear combination relationship can be estimated directly in the spectral space instead of in the RGB space. We show that this new regression approach outperforms the leading DNNs. More strikingly it—at least for current DNNs—questions the premise that these networks are carrying out meaningful patch-based computations. It is likely that current DNN models are unable to efficiently utilise the extra information brought by patch-based inputs, and this further implies that the available training data may not be enough to fully optimise their excessive model parameters (in our experiments, we have already selected one of the largest hyperspectral image datasets [5]).

Apart from our effort on improving regressions, we also propose two new problem definitions: exposure invariance and physical plausibility in SR. The problem of exposure invariance in SR entails how in reality when a ground-truth spectrum is scaled by a constant factor (i.e., getting brighter or dimmer), the resulting raw RGB camera response will also be scaled by the same factor, but a learned SR algorithm does not necessarily possess the same property.

The practical implication of this problem is that, when an SR algorithm works well on recovering spectra in the image region of a particular object, there is no promise that it will deliver the same performance if the object gets brighter or dimmer. Indeed, we found that all methods based on a non-linear mapping, including non-linear regressions and DNNs, degrade drastically when we brighten or dim the testing images.

Regarding exposure invariance, we made two contributions. First, we propose a new regression-based SR method which is a non-linear mapping function while possessing exposure invariance. Our second contribution is developing SR training frameworks that enforce exposure invariance on the SR algorithms. In the *chromaticity mapping* SR framework, we separate the input RGBs into two multiplying terms: chromaticitiy and brightness. While we recover a brightness-normalised spectrum from the chromaticity, the brightness term is kept constant and reapplied to the recovery that delivers the final spectral estimate. This approach ensures perfect exposure invariance, but the (potentially useful) brightness information is not utilised in training. Alternatively, in *data augmentation*, we apply random brightness scaling to the training data. This approach is shown to be preferred for DNN methods (compared to chromaticity mapping), but less effective for the regressions.

Then, we propose and investigate the physical plausibility issue of modern SR. Given a ground-truth spectrum, its corresponding RGB camera response can be calculated via numerically integrate the spectrum with the spectral sensitivities of the camera [93]. However, if we numerically integrate the algorithm-recovered spectrum with spectral sensitivities, we often arrive at a different RGB from the one suggested by the ground-truth, which means the recovered spectrum must not be the correct answer (i.e., the ground-truth). We now ask the following question: “Can we further improve the spectral accuracy of the existing SR methods via resolving this physical inconsistency?” We can confirm that the answer to this question is *yes*. We developed a universal post-processing step which not only can enforce physical plausibility to any algorithms, but also is mathematically proved to be able to improve the accuracy of every recovered spectrum.

In summary, this thesis presents the following contributions to the field of spectral reconstruction from RGB images:

- The regression-based SR methods are benchmarked against a leading DNN method, not only under the newer protocol the DNNs adopt, but also in our own designed extensive experiments including changing cameras and worst-case image set testing.
- We propose universal upgrades for the regression methods to optimise for a more similar loss metric compared to those used to evaluate and rank the recent DNN methods.
- An SR refining framework based on the dictionary-based sparse coding technique is proposed, which, when used in conjunction to the best-performing regression, delivers SR outperforming the leading DNN method.
- We point out a general problem that non-linear-mapping-based SR methods tend to significantly degrade when the exposure of the testing scenes changes, and propose several ways of enforcing performance invariance against exposure change when training an SR algorithm.
- We point out that most SR algorithms recover spectra that do not regenerate the input RGBs when applying with the underlying camera sensors' spectral sensitivities. We develop remedies for this issue, with one of our methods shown to be a universal improvement, in the sense that it is guaranteed to improve the spectral accuracy of any SR method.

1.1 Publications

Journal Publications

Lin, Y.-T., and Finlayson, G. D. A rehabilitation of pixel-based spectral reconstruction from RGB images. *Sensors 23*, 8 (2023), 4155.

Lin, Y.-T., and Finlayson, G. D. An investigation on worst-case spectral reconstruction from RGB images via Radiance Mondrian World assumption. *Color Research and Application 48*, 2 (2023), 230-242.

Lin, Y.-T., and Finlayson, G. D. On the optimization of regression-based spectral reconstruction. *Sensors 21*, 16 (2021), 5586.

Lin, Y.-T., and Finlayson, G. D. Physically plausible spectral reconstruction. *Sensors 20*, 21 (2020), 6399.

Conference Publications

Lin, Y.-T., and Finlayson, G. D. Evaluating the performance of different cameras for spectral reconstruction. In *Color and Imaging Conference* (2022), Society for Imaging Science and Technology, pp. 213–218.

Lin, Y.-T., and Finlayson, G. D. Investigating the upper-bound performance of sparse-coding-based spectral reconstruction from RGB images. In *Color and Imaging Conference* (2021), Society for Imaging Science and Technology, pp. 19–24.

Lin, Y.-T., and Finlayson, G. D. Recovering real-world spectra from RGB images under radiance mondrian-world assumption. In *International Colour Association (AIC) Conference* (2021), AIC, pp. 215–220.

Lin, Y.-T., and Finlayson, G. D. Reconstructing spectra from RGB images by relative error least-squares regression. In *Color and Imaging Conference* (2020), Society for Imaging Science and Technology, pp. 264–269.

Lin, Y.-T. Colour fidelity in spectral reconstruction from RGB images. In *London Imaging Meeting* (2020), Society for Imaging Science and Technology, pp. 144–148.

Lin, Y.-T., and Finlayson, G. D. Per-channel regularization for regression-based spectral reconstruction. In *Colour and Visual Symposium (CEUR Workshop Proceedings)* (2020), vol. 2688.

ARAD, B., TIMOFTE, R., BEN-SHAHAR, O., LIN, Y.-T., FINLAYSON, G. D., ET AL. NTIRE 2020 challenge on spectral reconstruction from an RGB image. In *Conference on Computer Vision and Pattern Recognition Workshops* (2020), IEEE, pp. 1806–1822.

LIN, Y.-T., AND FINLAYSON, G. D. Physically plausible spectral reconstruction from RGB images. In *Conference on Computer Vision and Pattern Recognition Workshops* (2020), IEEE, pp. 2257–2266.

LIN, Y.-T., AND FINLAYSON, G. D. Exposure invariance in spectral reconstruction from RGB images. In *Color and Imaging Conference* (2019), Society for Imaging Science and Technology, pp. 284–289.

Chapter 2

Literature Review

2.1 Imaging Technologies

2.1.1 Colour Imaging

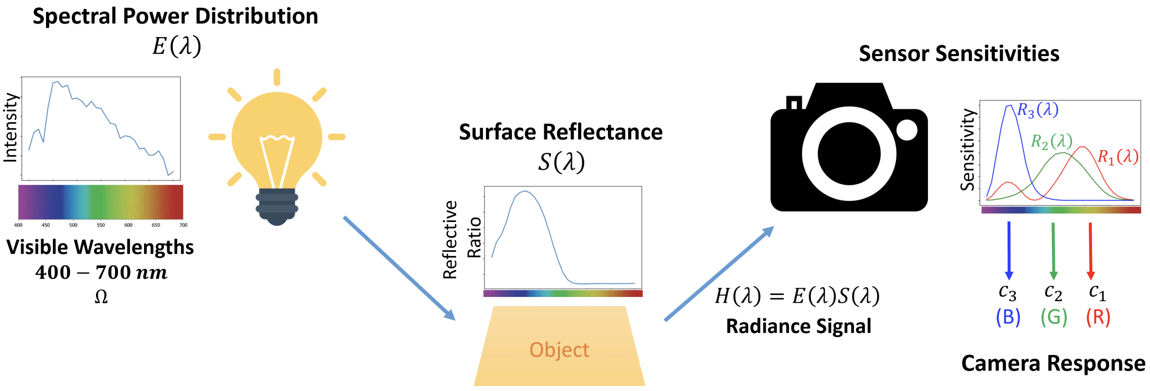


Figure 2.1: Illustration of colour image formation framework.

Summarised in Figure 2.1, the concept of *colour* (i.e., the "RGB") results from the interaction among the light source, object surfaces and a set of three colour sensors, especially on the *spectral* properties of these three factors. First, the light source emits electromagnetic (light) signals with varying intensities at different wavelengths, namely the Spectral Power Distribution (SPD), or simply the *illumination spectrum*. These signals are reflected from the object surfaces in the scene according to a spectrally varying ratio function called the *surface reflectance*. Importantly, it is well-known that surfaces made of the same material have unique surface reflectance, therefore the reflectance information is sometimes referred to as the "signature" of object surfaces [44]. Lastly, per pixel, an RGB camera captures the combined signals of illumination and object surfaces using three different sensors. In effect, each of these three types of sensors "weighted sums" the incoming signals into a single intensity output and with a different spectrally-varying sensitivity function, namely the *spectral sensitivity.*

Let us denote the illumination spectrum as $E(\lambda)$, the object surface's reflectance at a pixel as $S(\lambda)$, and the kth-sensor's spectral sensitivity function as $R_k(\lambda)$. A simple colour formation formula can be written as [93]:

$$\int_\Omega H(\lambda)R_k(\lambda)d\lambda = c_k \ ; \quad H(\lambda) = E(\lambda)S(\lambda) \ , \tag{2.1}$$

where $H(\lambda)$—called the *radiance*—is the combined signal of $E(\lambda)$ and $S(\lambda)$, representing the final signal that reaches the sensors. c_k is the resulting kth-sensor response ($k = 1, 2, 3$ respectively refer to as the R, G and B responses). Ω refers to the *visible range* of wavelength which roughly runs from 400 to 700 nanometers (nm). Since we human can only sense light within this range, the camera sensors are usually designed accordingly. In contrast, we note that both $E(\lambda)$ and $S(\lambda)$ (and therefore $H(\lambda)$) can have values outside this range, but those parts of the spectra do not influence the resulting colour responses c_k.

Though this colour imaging process mimics the colour vision of humans, it surely loses much information of the light. Indeed, while the radiance signal $H(\lambda)$ is a continuous function, in colour imaging we only record 3 numbers per pixels. This loss of information translates to the *ambiguity* of the 3-dimensional colour representation of light—there are *infinite* spectra that can plausibly produce the same RGB [29]—which leads to limiting performance for RGB cameras to be used in some computer vision and graphics applications.

To mitigate the limitation of RGB imaging, *multispectral* and *hyperspectral* cameras are designed to capture more information from the radiance signals, $H(\lambda)$.

2.1.2 Multispectral Imaging

In multispectral imaging, the captured channels per pixel increase from 3 in RGB imaging to around 5 to 10 channels. That is, we are using the same image formation formula as Equation (2.1) but with 5 to 10 different sensors. Though the spectral ambiguity still exists in this setup, multispectral imaging surely captures more spectral information than the RGB imaging.

Many designs are proposed for multispectral imaging. Inspired by the typical "bayer pattern" commonly used in RGB imaging [68] where a spatially repeating 2×2 pattern of RGB filters are used to cover a monochromatic sensor array, giving a single 3-channel RGB readings per 2×2 pixel neighbourhood, the design of Spectral Filter Array (SFA) [94] seeks to increase the number of different filters used per repeating unit pattern. Of course, to consider more than 4 spectral channels, the repeating pattern for SFA will need to be larger than 2×2. This means the spatial resolution of SFA can be much lower compared to what we usually get for an RGB colour camera.

Another common method is using a colour filter wheel [15], where several different filters are attached to a mechanical wheel which rotates at a controlled speed while each filter in turn covers the pupil of a monochromatic camera. By stacking the images captured by all the different filters, we get the multispectral measurement of the scene. However, this "spectral scanning" setup limits its temporal resolution, and as such it is not suitable for capturing objects that move.

Without giving up temporal and spatial resolutions, some works also propose to capture more than 3 channels by aligning multiple RGB cameras with different RGB spectral sensitivities to the exact same scene [63; 96]. Nonetheless, this setup requires exact image registration which is not always easy to do. Plus, with a system that includes multiple RGB cameras, we can expect this setup to be much bulkier than a regular RGB camera.

Compressive sensing is yet another approach to high-speed and high-resolution multispectral imaging. Including the use of coded aperture [10; 33; 34], faced reflectors [85], diffractive grating [55], digital micro-mirror device [74], and more recently the random printed mask [104], these approaches intertwine both the spectral and spatial information on the same image plane, following certain patterns to be decoded by specialised algorithms (many are based on machine learning). Nonetheless, failed decompression can alter the actual measured spectral information and/or leave uncompressed patterns on the resulting images.

Summarising the various multispectral imaging techniques introduced, we see that while being able to capture more of the spectral information than the RGB cameras, multispectral cameras face various challenges in terms of image quality, temporal resolution, and/or device bulkiness, when compared to our everyday RGB cameras.

2.1.3 Hyperspectral Imaging

In hyperspectral imaging, we are to measure the radiance signal, i.e., $H(\lambda)$ in Equation (2.1), at finely sampled wavelengths within the given range. Let us say we sample n equal-distanced points within the visible range $\Omega = [\lambda_1, \lambda_2, \cdots, \lambda_t, \cdots, \lambda_n]$. Then, the hyperspectral measurement at a pixel can be written as an n-dimensional vector:

$$\underline{\boldsymbol{h}} = [h_1, h_2, \cdots, h_t, \cdots, h_n]^{\mathsf{T}} \ ; \quad h_t = H(\lambda_t) = E(\lambda_t) S(\lambda_t) \ . \tag{2.2}$$

$\underline{\boldsymbol{h}}$ is often regarded as the ground-truth measurement of $H(\lambda)$ [7; 8].

In relation to RGB imaging, described in Equation (2.1), we can replace the continuous $H(\lambda)$ function with $\underline{\boldsymbol{h}}$—the measured radiance using a hyperspectral imaging device. Then, we may also discretise the spectral sensitivity functions, i.e., $R_k(\lambda)$; $k = 1, 2, 3$ in Equation

(2.1), so that it matches the wavelength samplings for $\underline{\boldsymbol{h}}$:

$$\mathbf{R} = \begin{bmatrix} R_1(\lambda_1) & R_1(\lambda_2) & \cdots & R_1(\lambda_t) & \cdots & R_1(\lambda_n) \\ R_2(\lambda_1) & R_2(\lambda_2) & \cdots & R_2(\lambda_t) & \cdots & R_2(\lambda_n) \\ R_3(\lambda_1) & R_3(\lambda_2) & \cdots & R_3(\lambda_t) & \cdots & R_3(\lambda_n) \end{bmatrix}^{\mathsf{T}} . \tag{2.3}$$

With those discretised representations, i.e., $\underline{\boldsymbol{h}}$ and $\mathbf{R}$, we can now approximate the integrals used in Equation (2.1) using inner products:

$$\mathbf{R}^{\mathsf{T}}\underline{\boldsymbol{h}} = \underline{\boldsymbol{c}} \ , \tag{2.4}$$

where $\underline{\boldsymbol{c}} = [c_1, c_2, c_3]^{\mathsf{T}} = [R, G, B]^{\mathsf{T}}$ is the 3-dimensional RGB colour vector [93]. In this thesis, we consider linearly independent columns for $\mathbf{R}$, referring to the standard tri-chromatic colour vision [16].

Equation (2.4) usefully describes how we can simulate the physically-accurate RGB response of an RGB camera using the hyperspectral measurement and the camera's spectral sensitivities. Also, with Equation (2.4) we see the power of hyperspectral imaging in colour applications. While the captured RGB colours differ from one camera to another, the hyperspectral measurement is device-independent, from which the colours of individual cameras can be derived directly, bypassing the usually-inevitable ill-posed conversions between devices [28].

In terms of the hyperspectral imaging techniques, for point measurement, spectral radiometer [76] is used, where the radiance measurement is spatially averaged over its field of view. To further capture radiance information with spatial content, a time-consuming scanning process is almost always required, including spatial scanning methods: "push broom" and "whisk broom" [48], and the spectral scanning methods using a Liquid Crystal Tunable Filter [108] or a prism-mask system [17]. Of course, these techniques require even longer time to capture one image compared to a multispectral device, which means it is almost impossible to use a hyperspectral device to capture moving objects. Furthermore, these devices are more expensive than multispectral devices, and even more so if we compare them with our everyday RGB cameras.

2.2 Spectral Reconstruction and Legacy Methods

The main topic of this thesis—Spectral Reconstruction (SR)—aims to find an algorithm that can reconstruct hyperspectral information from the RGB images. In effect, assuming we have a good enough SR-from-RGBs algorithm, we can capture hyperspectral information with an RGB camera (much more easily accessible, much less costly, and of much better spatial and temporal resolutions).

Let us denote an SR algorithm as a function $\Psi()$. We write:

$$\Psi(\underline{\boldsymbol{c}}_i) \approx \underline{\boldsymbol{h}}_i \ ; \quad \forall i \ , \tag{2.5}$$

where $\underline{\boldsymbol{h}}_i$ is the ith ground-truth radiance spectrum measured by a hyperspectral device, and $\underline{\boldsymbol{c}}_i$ is the RGB camera response of $\underline{\boldsymbol{h}}_i$. Here, i indexes the individual data points included in a considered dataset.

Assuming the hyperspectral device measures n values within the visible range, that is $\underline{\boldsymbol{h}}_i \in \mathbb{R}^n$, $\Psi()$ is then an $\mathbb{R}^3$ to $\mathbb{R}^n$ mapping ($n \gg 3$). Evidently, given an RGB, e.g., $\underline{\boldsymbol{c}}_i$, there are infinite possibilities in $\mathbb{R}^n$ that can be the answer. Yet, not all spectra in $\mathbb{R}^n$ can physically appear in the real world. The objective of $\Psi()$ is thus to find the best answer based on some preset criteria.

Below, we will start with reviewing some of the earliest SR methods. These methods seek physical clues to narrow down possible answers. Nevertheless, they can be inefficient and less accurate compared to methods that directly minimise spectral accuracy (which is the criterion most of the recent methods adopt). In addition, these early methods often consider the illumination spectrum separately from the surface reflectance, where the former is sometimes considered a fixed and known factor, and as such these SR methods seek to approximate the reflectance only:

$$\Psi(\underline{\boldsymbol{c}}_i; \underline{\boldsymbol{e}}) \approx \underline{\boldsymbol{s}}_i \ ; \quad \forall i \ , \tag{2.6}$$

where $\underline{\boldsymbol{e}}$ and $\underline{\boldsymbol{s}}_i$ are respectively the fixed illumination spectrum and surface reflectance of the ith data point.

2.2.1 3-D Linear Model

The earliest methods in the field of SR focus on finding a 3-dimensional representation of reflectance, and as such the RGB-to-reflectance mapping becomes well-posed [58; 2]. Usually, a linear model is used, where exactly 3 *basis vectors* are found such that all reflectances can be represented as linear combinations of these bases with minimal errors. In particular, the Principal Component Analysis (PCA), or equivalently the Singular Value Decomposition (SVD), are widely used to find the basis vectors [69].

Denoting the three (optimally found) basis vectors as $\underline{\boldsymbol{b}}^1$, $\underline{\boldsymbol{b}}^2$ and $\underline{\boldsymbol{b}}^3$, the reflectance spectrum, denoted as $\underline{\boldsymbol{s}}_i$, can be approximated by:

$$\underline{\boldsymbol{s}}_i \approx \sum_{j=1}^{3} \alpha_i^j \underline{\boldsymbol{b}}^j = \mathbf{B}^s \underline{\boldsymbol{\alpha}}_i \ ; \quad \forall i \ , \tag{2.7}$$

where α_i^j is the coefficient of the jth basis $\underline{\boldsymbol{b}}^j$ in the linear combination that approximates $\underline{\boldsymbol{s}}_i$, $\underline{\boldsymbol{\alpha}}_i = [\alpha_i^1, \alpha_i^2, \alpha_i^3]^{\mathsf{T}}$, and $\mathbf{B}^s = [\underline{\boldsymbol{b}}^1, \underline{\boldsymbol{b}}^2, \underline{\boldsymbol{b}}^3]$.

Returning to the mathematical relation between hyperspectral and colour measurements in Equation (2.4), and considering the assumed-fixed illumination term, $\underline{\boldsymbol{e}}$, separately from the varying surface reflectance in the database, $\underline{\boldsymbol{s}}_i$, we can write:

$$\underline{\boldsymbol{c}}_i = \mathbf{R}^{\mathsf{T}} \underline{\boldsymbol{h}}_i = \mathbf{R}^{\mathsf{T}} \text{diag}(\underline{\boldsymbol{e}}) \underline{\boldsymbol{s}}_i \ . \tag{2.8}$$

Then, we replace the reflectance vector $\underline{\boldsymbol{s}}_i$ by the linear model representation in Equation (2.7):

$$\underline{\boldsymbol{c}}_i \approx [\mathbf{R}^{\mathsf{T}}\text{diag}(\underline{\boldsymbol{e}})\mathbf{B}^s]\underline{\boldsymbol{\alpha}}_i = \boldsymbol{\Lambda}\underline{\boldsymbol{\alpha}}_i \ , \tag{2.9}$$

where $\boldsymbol{\Lambda}$ is a *known* 3×3 non-singular matrix provided that all columns of $\mathbf{R}$ are linearly independent (i.e., standard tri-chromatic vision), and all $\mathbf{R}$, $\underline{\boldsymbol{e}}$ and $\mathbf{B}^s$ are fixed. $\boldsymbol{\Lambda}$ is sometimes referred to as the *lighting matrix* [60] that is, in essence, a matrix operator combining the effects of illumination and sensor, which acts on the reflectances to derive their corresponding colours.

Since $\boldsymbol{\Lambda}$ is a known and non-singular squared matrix, an equivalent relation between $\underline{\boldsymbol{\alpha}}_i$ and $\underline{\boldsymbol{c}}_i$ can be written as:

$$\underline{\boldsymbol{\alpha}}_i \approx \boldsymbol{\Lambda}^{-1}\underline{\boldsymbol{c}}_i \ ; \quad \forall i \ . \tag{2.10}$$

By combining Equation (2.7) and (2.10), we get:

$$\underline{\boldsymbol{s}}_i \approx \mathbf{B}^s\underline{\boldsymbol{\alpha}}_i \approx \mathbf{B}^s\boldsymbol{\Lambda}^{-1}\underline{\boldsymbol{c}}_i \ ; \quad \forall i \ , \tag{2.11}$$

which solves the RGB-to-reflectance mapping problem (Equation (2.6)).

Of course, the main problem of this approach is its 3-dimensional assumption for the reflectance. As suggested in later studies, such as [41; 72; 51; 59; 64], at least a 5- to 8-dimensional linear model is required to sufficiently describe the surface reflectances in real-world datasets. Still, the idea of pursuing effective linear models to represent spectral data remains important for later studies and also some of the new contents delivered in this thesis.

2.2.2 Bayesian Inference

Brainard and Freeman [14] proposed a Bayesian inference model to solve SR. Usefully, their formulation is applicable to any arbitrary m-dimensional linear model of reflectance ($m \leq n$, where n is the actual dimension of the reflectances), though in the original publication, $m =$

3 is used. Additionally, they assume varying illumination spectrum, denoted as $\underline{\boldsymbol{e}}_i$, which is to be estimated from the RGB, $\underline{\boldsymbol{c}}_i$, alongside the recovery of the surface reflectance $\underline{\boldsymbol{s}}_i$.

Here, for every radiance spectrum we wish to recover, we are to recover $\underline{\boldsymbol{e}}_i$ and $\underline{\boldsymbol{s}}_i$ separately. Using respective datasets, we can optimise separate m-dimensional linear models that optimally represent both spectral components:

$$\begin{cases} \underline{\boldsymbol{e}}_i \approx \mathbf{B}^e \underline{\boldsymbol{\alpha}}_i^e \\ \underline{\boldsymbol{s}}_i \approx \mathbf{B}^s \underline{\boldsymbol{\alpha}}_i^s \end{cases} ; \quad \forall i \ , \tag{2.12}$$

where $\mathbf{B}^e$ and $\mathbf{B}^s$ are $n \times m$ matrices of bases, and $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$ are m-vectors of coefficients. Then, SR in this case wishes to recover $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$ from their RGB observation $\underline{\boldsymbol{c}}_i$ which is related to $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$ by:

$$\underline{\boldsymbol{c}}_i \approx \mathbf{R}^{\mathsf{T}} \mathrm{diag}(\mathbf{B}^e \underline{\boldsymbol{\alpha}}_i^e) \mathbf{B}^s \underline{\boldsymbol{\alpha}}_i^s \ . \tag{2.13}$$

This approximation is derived from Equation (2.9) where we substitute the fixed illumination spectrum $\underline{\boldsymbol{e}}$ with the varying $\underline{\boldsymbol{e}}_i$ represented by Equation (2.12).

In a Bayesian sense, we are to formulate the posterior distribution of $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$, denoted as $p(\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s | \underline{\boldsymbol{c}}_i)$, following the Bayes' rule:

$$p(\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s | \underline{\boldsymbol{c}}_i) = \frac{p(\underline{\boldsymbol{c}}_i | \underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s) p(\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s)}{p(\underline{\boldsymbol{c}}_i)} \ . \tag{2.14}$$

Our target is then to find the maximal value in $p(\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s | \underline{\boldsymbol{c}}_i)$ that suggests the estimation of $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$.

Let us further investigate each term on the right hand side of Equation (2.14). First, the colour's probability distribution, $p(\underline{\boldsymbol{c}}_i)$, is independent to the estimation targets: $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$, thus it is regarded as a constant. Next, the $p(\underline{\boldsymbol{c}}_i | \underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s)$ term entails how likely that $\underline{\boldsymbol{c}}_i$ is the colour when $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$ are respectively the illumination spectrum and surface reflectance (or their linear model coefficients thereof). From Equation (2.13), we know that the colour is *decided* when both the illumination and reflectance are given. Therefore, ignoring noise,

$p(\underline{\boldsymbol{c}}_i|\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s) = 1$ when $\underline{\boldsymbol{c}}_i$ is the exact colour derived from $\underline{\boldsymbol{\alpha}}_i^e$ and $\underline{\boldsymbol{\alpha}}_i^s$, and $p(\underline{\boldsymbol{c}}_i|\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s) = 0$ elsewhere. I.e., $p(\underline{\boldsymbol{c}}_i|\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s)$ is known as a *delta function.* We note that Brainard and Freeman [14] also consider the $p(\underline{\boldsymbol{c}}_i|\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s)$ formulation in a noisy case. Interested readers are pointed to the original paper for more information.

Finally, $p(\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s)$ is called the *prior* distribution of illumination and surface reflectance. Assuming that the reflectances do not change under different illuminations, the joint prior distribution of the two factors becomes the product of the two individual prior distributions:

$$p(\underline{\boldsymbol{\alpha}}_i^e, \underline{\boldsymbol{\alpha}}_i^s) = p(\underline{\boldsymbol{\alpha}}_i^e)p(\underline{\boldsymbol{\alpha}}_i^s) \ . \tag{2.15}$$

Respectively, $p(\underline{\boldsymbol{\alpha}}_i^e)$ and $p(\underline{\boldsymbol{\alpha}}_i^s)$ are modeled by truncated normal distributions:

$$p(\underline{\boldsymbol{\alpha}}_i^e) = \begin{cases} \mathcal{N}(\underline{\boldsymbol{\mu}}^e, \boldsymbol{\Sigma}^e) & \text{all components in } \mathbf{B}^e\underline{\boldsymbol{\alpha}}_i^e \geq 0 \\ 0 & \text{otherwise} \end{cases} , \tag{2.16}$$

and

$$p(\underline{\boldsymbol{\alpha}}_i^s) = \begin{cases} \mathcal{N}(\underline{\boldsymbol{\mu}}^s, \boldsymbol{\Sigma}^s) & \text{all components in } \mathbf{B}^s\underline{\boldsymbol{\alpha}}_i^s \geq 0 \\ 0 & \text{otherwise} \end{cases} . \tag{2.17}$$

Here, $\mathcal{N}$ represents the normal distribution, $\underline{\boldsymbol{\mu}}^e$ and $\boldsymbol{\Sigma}^e$ are the mean and covariance matrix of $\underline{\boldsymbol{\alpha}}_i^e$ in the considered illumination dataset, and likewise for $\underline{\boldsymbol{\mu}}^s$ and $\boldsymbol{\Sigma}^s$ in the reflectance dataset. The physical constraints—the conditions to prevent negative values for the derived illumination and reflectance—are placed so that the final estimations are physically realisable in the real world (i.e., physical spectral values cannot be negative). With the derivations from Equation (2.14) to (2.17), we solve SR by finding the maximal point of the posterior distribution.

A general advantage of Bayesian approach is that it not only returns the most probable estimate, but also shows the probability for other feasible candidates to be the correct estimate. However, the Bayesian's inference process is known for being very slow to run, especially if we wish to consider this method beyond 3-dimensional linear models (i.e., $m > 3$).

2.2.3 Bayesian with Metamer Sets Constraint

While we anticipate to use a 5- to 8-dimensional representation of reflectances for SR, Morovic and Finlayson [60] show that, with known camera's sensor spectral sensitivities, we can already fix 3 dimensions of the linear model given each input RGB, namely the *metamer sets* constraint. Advantageously, this means that we have 3 dimensions less information to estimate, which can lead to faster inference. Moreover, the uncertainty of SR is further bounded by this new constraint.

Let us represent the n-dimensional surface reflectances using an m-dimensional linear model: $\underline{\boldsymbol{s}}_i \approx \mathbf{B}^s \underline{\boldsymbol{\alpha}}_i$, $\forall i$, where $\mathbf{B}^s$ is $n \times m$ whose columns are individual basis vectors, and $\underline{\boldsymbol{\alpha}}_i$ is an m-component coefficient vector uniquely representing $\underline{\boldsymbol{s}}_i$. Then, the corresponding colour of a reflectance under a fixed known illumination, denoted as $\underline{\boldsymbol{c}}_i$, can be derived using Equation (2.9), that is

$$\underline{\boldsymbol{c}}_i = \boldsymbol{\Lambda} \underline{\boldsymbol{\alpha}}_i \ , \tag{2.18}$$

only that here the dimension of the lighting matrix, $\boldsymbol{\Lambda}$, is $3 \times m$ (because here an m-dimensional linear model is used instead of the 3-dimensional model used previously in Section 2.2.1).

With respect to $\boldsymbol{\Lambda}$, we can separate $\underline{\boldsymbol{\alpha}}_i$ into two components, $\underline{\boldsymbol{\alpha}}_i^f$ and $\underline{\boldsymbol{\alpha}}_i^b$, such that:

$$\underline{\boldsymbol{\alpha}}_i = \underline{\boldsymbol{\alpha}}_i^f + \underline{\boldsymbol{\alpha}}_i^b \ ; \quad \text{subject to} \begin{cases} \boldsymbol{\Lambda}\underline{\boldsymbol{\alpha}}_i^f = \underline{\boldsymbol{c}}_i \\ \boldsymbol{\Lambda}\underline{\boldsymbol{\alpha}}_i^b = \underline{\mathbf{0}} \end{cases} ; \quad \forall i \ . \tag{2.19}$$

Here, the $\underline{\boldsymbol{\alpha}}_i^f$ component is called the *fundamental metamer* of $\underline{\boldsymbol{\alpha}}_i$, which is the only component out of the two contributes to the colour image formation (i.e., when multiplying the

$\mathbf{\Lambda}$ matrix we derive the colour vector $\underline{\boldsymbol{c}}_i$). On the other hand, $\underline{\boldsymbol{\alpha}}_i^b$ is called the *metameric black* which, when multiplying $\mathbf{\Lambda}$, returns a 3-dimensional zero vector $\underline{\mathbf{0}}$ [29].

In a linear algebraic point of view, $\underline{\boldsymbol{\alpha}}_i^f$ is $\underline{\boldsymbol{\alpha}}_i$ projected onto the 3-dimensional vector space spanned by the row vectors of $\mathbf{\Lambda}$ (i.e., the *row space* of $\mathbf{\Lambda}$, or $\mathcal{C}(\mathbf{\Lambda}^\mathsf{T})$), whereas $\underline{\boldsymbol{\alpha}}_i^b$ lies in the $(m-3)$-dimensional *null-space* of $\mathcal{C}(\mathbf{\Lambda}^\mathsf{T})$. Combined, the two spaces form the $\mathbb{R}^m$ space where $\underline{\boldsymbol{\alpha}}_i$ lies in. Based on this insight, we derive the respective *projection matrices* [83] which usefully derive $\underline{\boldsymbol{\alpha}}_i^f$ and $\underline{\boldsymbol{\alpha}}_i^b$ from $\underline{\boldsymbol{\alpha}}_i$:

$$\begin{cases} \mathbf{P}^f = \mathbf{\Lambda}^\mathsf{T}(\mathbf{\Lambda}\mathbf{\Lambda}^\mathsf{T})^{-1}\mathbf{\Lambda} & , \quad \text{such that } \mathbf{P}^f\underline{\boldsymbol{\alpha}}_i = \underline{\boldsymbol{\alpha}}_i^f \ , \\ \mathbf{P}^b = \mathbf{I} - \mathbf{P}^f & , \quad \text{such that } \mathbf{P}^b\underline{\boldsymbol{\alpha}}_i = \underline{\boldsymbol{\alpha}}_i^b \ . \end{cases} \tag{2.20}$$

In terms of spectral reconstruction, we are to recover $\underline{\boldsymbol{\alpha}}_i$ given an input colour $\underline{\boldsymbol{c}}_i$. Or, equivalently, we can recover $\underline{\boldsymbol{\alpha}}_i^f$ and $\underline{\boldsymbol{\alpha}}_i^b$ separately from $\underline{\boldsymbol{c}}_i$. First, let us focus on the $\underline{\boldsymbol{\alpha}}_i^f$ term. According to the top line of Equation (2.20), we write:

$$\underline{\boldsymbol{\alpha}}_i^f = \mathbf{P}^f\underline{\boldsymbol{\alpha}}_i = \mathbf{\Lambda}^\mathsf{T}(\mathbf{\Lambda}\mathbf{\Lambda}^\mathsf{T})^{-1}[\mathbf{\Lambda}\underline{\boldsymbol{\alpha}}_i] \ . \tag{2.21}$$

Given that $\mathbf{\Lambda}\underline{\boldsymbol{\alpha}}_i = \underline{\boldsymbol{c}}_i$ is a known physical relation between $\underline{\boldsymbol{\alpha}}_i$ and $\underline{\boldsymbol{c}}_i$ (see Equation (2.18)), we get:

$$\underline{\boldsymbol{\alpha}}_i^f = \mathbf{\Lambda}^\mathsf{T}(\mathbf{\Lambda}\mathbf{\Lambda}^\mathsf{T})^{-1}\underline{\boldsymbol{c}}_i \ . \tag{2.22}$$

Evidently, the $\underline{\boldsymbol{\alpha}}_i^f$ component is *unique* to the given RGB value $\underline{\boldsymbol{c}}_i$. That is, all reflectances whose colour is $\underline{\boldsymbol{c}}_i$ have exactly the same fundamental metamer component which is $\underline{\boldsymbol{\alpha}}_i^f$. In addition, Equation (2.22) derives $\underline{\boldsymbol{\alpha}}_i^f$ from $\underline{\boldsymbol{c}}_i$ directly without the need for any additional estimation.

This leaves the spectral reconstruction problem only to estimate the $\underline{\boldsymbol{\alpha}}_i^b$ metameric black component from $\underline{\boldsymbol{c}}_i$. Unlike $\underline{\boldsymbol{\alpha}}_i^f$ which is fixed upon given the input RGB $\underline{\boldsymbol{c}}_i$, $\underline{\boldsymbol{\alpha}}_i^b$ can be any vector lies in the $(m-3)$-dimensional null-space of $\mathcal{C}(\mathbf{\Lambda}^\mathsf{T})$. We can write $\underline{\boldsymbol{\alpha}}_i^b$ in another bases-coefficients representation:

$$\underline{\boldsymbol{\alpha}}_i^b = \mathbf{B}^b \underline{\boldsymbol{\beta}}_i^b \ , \tag{2.23}$$

where $\underline{\boldsymbol{\beta}}_i^b$ is an $(m-3)$-vector of coefficients, and $\mathbf{B}^b$ is an $m \times (m-3)$ dimensional matrix whose columns are the basis vectors of the null-space of $\mathcal{C}(\boldsymbol{\Lambda}^{\mathsf{T}})$. These basis vectors can be derived from the projection matrix $\mathbf{P}^b$ in the bottom Equation (2.20) as the $m-3$ linearly independent columns of $\mathbf{P}^b$ [83], which can be found using, e.g., the Gram–Schmidt orthogonalisation procedure [22].

In effect, given an input RGB, $\underline{\boldsymbol{c}}_i$, we are to find the most probable estimate out of all reflectances whose fundamental metamer component is exactly $\underline{\boldsymbol{\alpha}}_i^f$ while only differ with each other in their metameric black component. In the parlance of Morovic and Finlayson [60], these reflectances form a *metamer set*. Mathematically, we write:

$$\mathcal{M}(\underline{\boldsymbol{c}}_i; \boldsymbol{\Lambda}) = \left\{ \underline{\boldsymbol{\alpha}}_i^f + \mathbf{B}^b \underline{\boldsymbol{\beta}}_i^b \ \middle| \ \underline{\boldsymbol{\beta}}_i^b \in \mathbb{R}^{m-3} \right\} \ , \tag{2.24}$$

where $\mathcal{M}$ denotes the metamer set with respect to the input RGB $\underline{\boldsymbol{c}}_i$ and the fixed known $\boldsymbol{\Lambda}$ matrix—the two factors needed to derive the $\underline{\boldsymbol{\alpha}}_i^f$ component (Equation (2.22)).

Adopting a Bayesian inference process (Section 2.2.2), analogously, a Gaussian distribution is used to model the prior distribution of $\underline{\boldsymbol{\alpha}}_i$. Then, on input a particular $\underline{\boldsymbol{c}}_i$, we are to calculate the corresponding $\mathcal{M}(\underline{\boldsymbol{c}}_i; \boldsymbol{\Lambda})$ and *intersect* $\mathcal{M}$ with the prior distribution of $\underline{\boldsymbol{\alpha}}_i$ before solving the most probable estimation within this intersecting region.

Effectively, we are solving SR such that all recoveries lie in their respective metamer sets derived from their corresponding RGBs. This ensures that the colours of all recovered reflectances coincide the input RGBs. This idea will be a core insight we use to develop our physically plausible SR solutions presented in Chapter 7.

Note that in the original work of Morovic and Finlayson [60], there are two other physical constraints used, namely the physical realisability and naturalness constraints, which further bound the uncertainty of their SR solution. Interested readers are pointed to the original paper for more details.

2.3 Regression

Unlike the methods presented in the previous section where explicit physical constraints are used, in regression we focus on statistically minimising the errors between the ground-truth and reconstructed spectra. In addition, regression can be used to estimate the full n-dimensional spectrum directly and effectively, without the need for a lower-dimensional linear model. Here, we consider radiance reconstruction using regression (as opposed to reflectance reconstruction discussed in the legacy methods).

2.3.1 Formulations

In Linear Regression (LR), simply, an $n\times 3$ linear regression matrix $\mathbf{M}$ (n is the dimension of the spectra) is used to predict all spectra in a dataset from their corresponding RGBs:

$$\mathbf{M}\underline{\boldsymbol{c}}_i \approx \underline{\boldsymbol{h}}_i \;\; ; \;\; \forall i \; , \tag{2.25}$$

where i indexes individual data points, and $\underline{\boldsymbol{c}}_i$ and $\underline{\boldsymbol{h}}_i$ are respectively the RGB and the ground-truth spectrum of the ith data point. Or, equivalently:

$$\mathbf{M}\mathbf{C} \approx \mathbf{H} \; , \tag{2.26}$$

where all matching RGB and spectral data is arranged into the matching columns of $\mathbf{C}$ and $\mathbf{H}$, respectively. Given N as the number of data points of concern (training or testing), the matrix dimensions of $\mathbf{C}$ and $\mathbf{H}$ are $3\times N$ and $n\times N$, respectively. In effect, the elements in $\mathbf{M}$ are the only trainable parameters used for the RGB-to-spectrum mapping in LR.

To further improve the performance of LR, the Polynomial Regression (PR) method seeks to introduce non-linearity to the mapping, by applying a fixed-order polynomial expansion, denoted as a function $\varphi()$, to all the RGBs before mapping them to spectra using a linear transformation, i.e.:

$$\begin{cases} \mathbf{M}\varphi(\underline{\boldsymbol{c}}_i) \approx \underline{\boldsymbol{h}}_i \ \ ; \ \ \forall i \ , \\ \mathbf{M}\mathbf{C}^{\varphi} \approx \mathbf{H} \ , \end{cases} \tag{2.27}$$

where the $^{\varphi}$ superscript of $\mathbf{C}^{\varphi}$ indicates that its columns are polynomial expansions of the RGBs instead of the RGBs in the columns of $\mathbf{C}$. Here, given $\underline{\boldsymbol{c}}_i = [R, G, B]^{\mathsf{T}}$, the 2nd-, 3rd-, and 4th-order polynomial expansions are written as follows:

$$\begin{aligned}
\text{2nd-order: } \varphi(\underline{\boldsymbol{c}}_i) \ &= \ \Big[R, G, B, R^2, G^2, B^2, RG, GB, RB\Big]^{\mathsf{T}} \\
\text{3rd-order: } \varphi(\underline{\boldsymbol{c}}_i) \ &= \ \Big[R, G, B, R^2, G^2, B^2, RG, GB, RB, \\
&\qquad R^3, G^3, B^3, RG^2, GB^2, RB^2, GR^2, BG^2, BR^2, RGB\Big]^{\mathsf{T}} \\
\text{4th-order: } \varphi(\underline{\boldsymbol{c}}_i) \ &= \ \Big[R, G, B, R^2, G^2, B^2, RG, GB, RB, \\
&\qquad R^3, G^3, B^3, RG^2, GB^2, RB^2, GR^2, BG^2, BR^2, RGB, \\
&\qquad R^4, G^4, B^4, R^3G, R^3B, G^3R, G^3B, B^3R, B^3G, \\
&\qquad R^2G^2, G^2B^2, R^2B^2, R^2GB, G^2RB, B^2RG\Big]^{\mathsf{T}} \ .
\end{aligned} \tag{2.28}$$

Of course, with PR, and especially with a higher-order polynomial expansion adopted as $\varphi()$, more complex mapping function can be achieved. For example, if we use the 2nd-order expansion for PR, the length of polynomial expansion, $\varphi(\underline{\boldsymbol{c}}_i)$, is 9, and subsequently the dimension of the regression matrix $\mathbf{M}$ increases from $n \times 3$ for LR to $n \times 9$—that is, 3 times more mapping parameters are used in 2nd-order PR compared to LR. Nevertheless, a more complex mapping does not always suggest better SR performance, mainly because it increases the risk of *overfitting*, that is the trained mapping works well on training data but fails to generalise to the unseen testing data [98]. The overfitting problem will be introduced and discussed in more details in the next sections.

2.3.2 Solving Least-Squares Regressions

Now let us consider how we solve for the regression matrix $\mathbf{M}$ in LR and PR. In essence, we must define what we meant by the $\approx$ symbols in Equation (2.26) and (2.27). Most commonly, we seek to minimise the "sum-of-squares" between the estimations and ground-truths:

$$\min_{\mathbf{M}} \ \ ||\mathbf{MC}-\mathbf{H}||_2^2 \ . \tag{2.29}$$

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. Moreover, we drop the $^{\varphi}$ superscript of $\mathbf{C}^{\varphi}$ used in Equation (2.27) for simplicity. That is, in the following derivations the columns of $\mathbf{C}$ can mean the RGBs for LR, or the polynomial-expanded RGB features for PR.

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

$$\mathbf{M} = \mathbf{HC}^{\dagger} = \mathbf{HC}^{\mathsf{T}}[\mathbf{CC}^{\mathsf{T}}]^{-1} \ , \tag{2.30}$$

where † represents the Moore-Penrose inverse operation [66].

2.3.3 The Overfitting Problem and Regularised Least Squares

In Equation (2.29) and (2.30), the data included in the columns of $\mathbf{C}$ and $\mathbf{H}$ is called the *training* data. In practice, we wish to train the regressions with a fixed set of training data, while this trained mapping is to be used on some unseen query data. Nevertheless, the regression matrix $\mathbf{M}$ obtained from Equation (2.30) only ensures the minimisation of errors within the training data. For a set of unseen RGB and spectral data, denoted as $\mathbf{C}'$ and $\mathbf{H}'$, it is *a priori* possible that while $||\mathbf{MC}-\mathbf{H}||$ is small but $||\mathbf{MC}'-\mathbf{H}'||$ is, comparatively, very large. This problem is so-called "overfitting" [98; 87].

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 set $\mathbf{H}' = \mathbf{H}$ (directly use the training spectra as testing data), but now we recover $\mathbf{H}$ from slightly perturbed training RGBs: $\mathbf{C}' = \mathbf{C} + \boldsymbol{\epsilon}$. Here, $\boldsymbol{\epsilon}$ 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 $||\mathbf{M}[\mathbf{C} + \boldsymbol{\epsilon}] - \mathbf{H}|| \gg ||\mathbf{MC} - \mathbf{H}||$. That is, it fails to plausibly recover the spectra in the training set, $\mathbf{H}$, even though they are exactly the ones used in training.

Another facet of this problem is that if we actually attempt to find the best regression matrix for $[\mathbf{C} + \boldsymbol{\epsilon}]$ (the noisy RGB training data), 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 regularisation (a.k.a. Tikhonov regularisation) [87] is often incorporated when training a regression. While solving the minimisation in Equation (2.29), we add another penalty term which bounds the magnitude of the regression matrix $\mathbf{M}$:

$$\min_{\mathbf{M}} \; ||\mathbf{MC} - \mathbf{H}||_2^2 + \gamma ||\mathbf{M}||_2^2 \; , \tag{2.31}$$

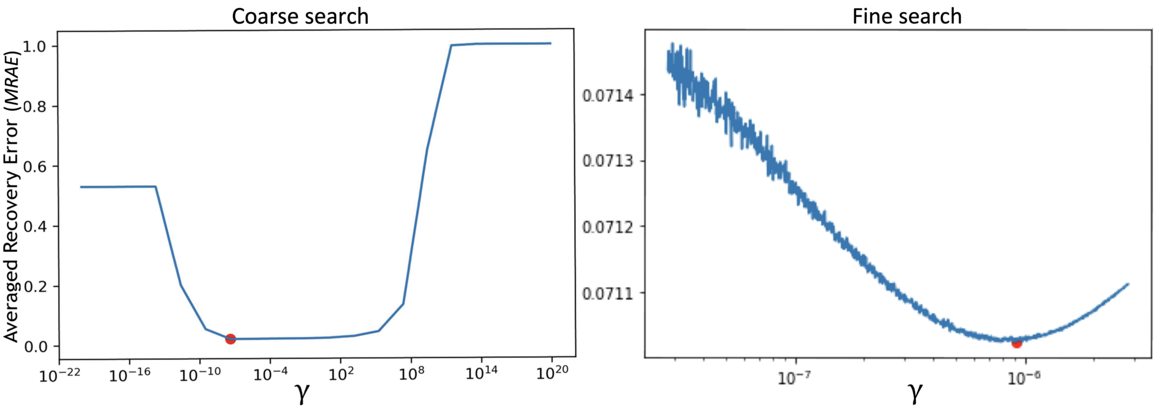
where the solution of $\mathbf{M}$ can still be written in closed form [42]:

$$\mathbf{M} = \mathbf{HC}^{\mathsf{T}}[\mathbf{CC}^{\mathsf{T}} + \gamma \mathbf{I}]^{-1} \; . \tag{2.32}$$

Here, the γ parameter—which is set by the user—controls how $||\mathbf{M}||_2^2$ mitigates the minimisation of the sum-of-squares fitting error, and the $\mathbf{I}$ matrix is the identity matrix with a dimension of 3×3 for Linear Regression (LR) and $p \times p$ for Polynomial Regression (PR) with p being the length of the polynomial expansion ($\varphi(\underline{\boldsymbol{c}})$ in Equation (2.27)).

When the solved $\mathbf{M}$ has a bounded norm, the regression has the stability property we desire. That is, if we perturb $\mathbf{C}$ (i.e., the training RGB data) 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 [87] for a fuller discussion of how ridge regularisation 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. Typically, a cross-validation parameter selection methodology [31] is used, where 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 set of *validation* data, which are then evaluated using the desired evaluation metric. In our experiment, a U-shaped curve like Figure 2.2(a) is usually obtained when plotting the averaged validation error against γ, where the minimal point (the red dot in the plot) indicates the selected parameter.



(a) (b)

Figure 2.2: Illustrations of searching for the optimal regularisation parameter. The averaged recovery error (vertical axis) is calculated over the validation data (a separate data set that is not the training data). The red dot in each graph indicates the minimal error and the suggested regularisation parameter. (a) Coarse search in a wide range of γ. (b) Fine search around the red dot in graph (a).

Let us examine Figure 2.2(a) in more detail. On the left, we see a plateau under roughly $\gamma = 10^{-16}$. This is caused by the floating-point precision in our calculations (under this level the selection of different γ's will not make a difference in calculation). Then, we look at the far right of the curve. As γ becomes too large, the need to solve for an $\mathbf{M}$ that has a small (bounded) norm becomes imperative, which ultimately renders all numbers in $\mathbf{M}$ closed to zero (and since the MRAE error metric used here is calculated relative to the ground-truth values, with predictions approaching 0 in all values, the error approaches 1, i.e., 100% error; MRAE will be introduced later in Section 3.1.4).

Lastly, we see that the optimal point in Figure 2.2(a) falls in the valley in the middle. The implication of this apparently broad and flat valley is that the changes in error here are at a much smaller order of magnitude compared to the error levels in overfitting and over-regularised cases. Indeed, as we conduct a finer search around the minimal point of Figure 2.2(a), we can further discern the minimal point among the gradually changing trends in the valley, as shown in Figure 2.2(b).

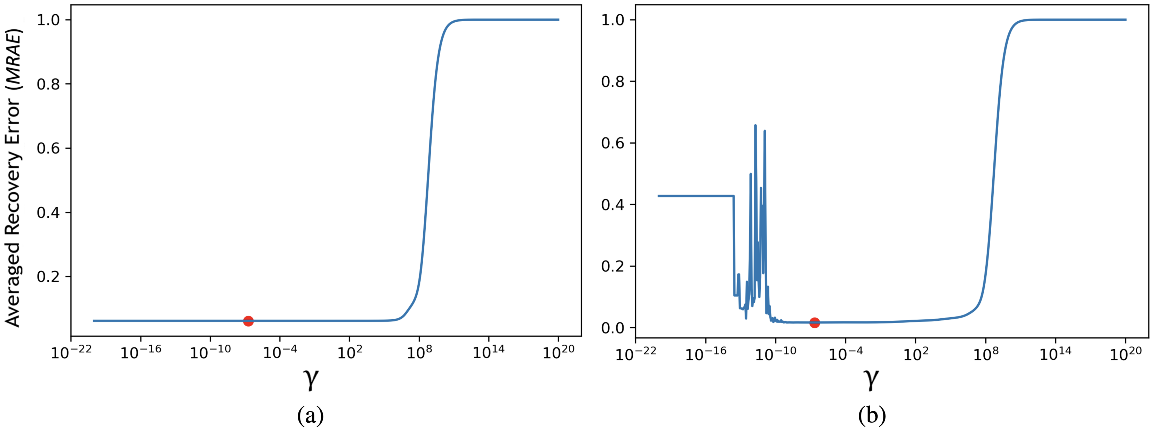


Figure 2.3: Illustrations of other possible validation curves when searching for the optimal regularisation parameter.

Some other possible shapes for the validation curves are shown in Figure 2.3. Figure 2.3(a) shows an example when the problem of overfitting is less significant (a small γ does not incur significantly large error). There are several possible occasions for this to happen, including when the training data represents the validation data well, when the model is too simple for the data, and so on [52]. Then, we see Figure 2.3(b). Here, some noisy results appear on the overfitting side of the curve (i.e., when γ is small). This indicates that the considered model (typically a more non-linear/heavier-parameterised model) is very unstable when it is overfitted (a small change in γ value causes big differences in predicted values). While we do not rule out the existence of other possible validation curve structures, we note that the general goal of regularisation via cross validation is to find the lowest point of the validation-error-versus-γ curve.

Although the ridge regularisation 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 regularisation approaches used in the literature, including the LASSO [86] (where the ℓ_1 penalty term is used) and Elastic Net approach [106] (where both ℓ_1 and ℓ_2 penalty terms are added). It is known that LASSO provides more robust regularisation compared to the ℓ_2 ridge regularisation, 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 [86] and [106], respectively.

2.4 Clustering-Based Methods

In regression, a global mapping matrix is used for all training and unseen query data (Section 2.3.1). With the amount of publicly available spectral data increases from discrete point measurements to hyperspectral "images" (where the spectral measurement at each pixel provides a data point), the methods I am going to introduce in this section use *clustering* techniques to help the optimisation of the mapping becoming more localised in individual RGB regions.

2.4.1 Radial-Basis-Function Network

In Radial-Basis-Function Network (RBFN) [62], the K-means algorithm [57] is used to cluster all RGB training data into K clusters, and, accordingly, we record the K cluster *centres* in an RGB dictionary $\mathbf{D}^c$:

$$\mathbf{D}^c = K\text{-means}(\mathbf{C}) = \left[\underline{\boldsymbol{c}}^1, \underline{\boldsymbol{c}}^2, \cdots, \underline{\boldsymbol{c}}^j, \cdots, \underline{\boldsymbol{c}}^K\right] \ . \tag{2.33}$$

Here, $\mathbf{C}$ represents the data matrix of the training-set RGBs.

On input of a training or query RGB, denoted as $\underline{\boldsymbol{c}}_i$, we write the radial-basis-function feature vector as:

$$\varphi(\underline{\boldsymbol{c}}_i) = \Big[1, \phi(||\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^1||), \phi(||\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^2||), \cdots, \phi(||\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^j||), \cdots, \phi(||\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^K||)\Big]^{\mathsf{T}} , \tag{2.34}$$

where the ℓ_2 distance between $\underline{\boldsymbol{c}}_i$ and each centre recorded in $\mathbf{D}^c$ is calculated, and, taken those distances individually as input, $\phi : \mathbb{R} \mapsto \mathbb{R}$ is a fixed non-linear function. Examples of the functions used as the ϕ function in the literature are presented in Chen et al. [20]. Commonly, a Gaussian radial-basis-function is used. Taking the jth centre $\underline{\boldsymbol{c}}^j$ as an example, we write:

$$\phi(||\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^j||) = \exp\left(- \frac{||\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^j||^2}{2\sigma^2} \right) . \tag{2.35}$$

Here, a fixed Gaussian's width factor $\sigma = \frac{d_{max}}{\sqrt{2K}}$ can be used, where d_{max} is the maximal distance the input colour $\underline{\boldsymbol{c}}_i$ has among the K cluster centres [12].

Then, with the φ features calculated for all colours in the training data set, we solve a single regression matrix $\mathbf{M}$ (with a dimension of $n \times (K+1)$ where n is the length of the spectral vectors) that maps these features to spectral estimates with minimal errors. I.e., we follow the same global regression optimisation process as Polynomial Regression (PR; Equation (2.27)), while here we use the radial-basis-function feature defined in Equation (2.34) instead of the polynomial expansion used in PR.

Effectively, RBFN defines a mapping structure equivalent to a single-layer neural network (specifically using radial-basis-functions as activation functions for the neurons) [20], while being able to solve for a least-squares minimum in closed form.

2.4.2 Sparse Coding

The basic assumption behind sparse coding approach is that all spectra can either be found in or as linear combinations of fewer spectra. It is assumed that the designated continuous data distribution can be effectively approximated by linear interpolations of a set of *representative spectra* found by algorithms such as K-SVD [3] (which is a generalisation of K-means). More specifically, in K-SVD, representatives are found such that all data points can be derived as linear combinations of the representatives with minimal errors. There is also an adjustable integer factor called *sparsity*, denoted as ℓ, which further constrains that only ℓ out of all representatives will be used in each linear combination.

Two main sparse coding approaches commonly benchmarked in the literature of SR are Arad and Ben-Shahar [5] and Aeschbacher et al. [1], where the latter is equivalent to a *local linear regression* approach. Usefully, with a regression formulation, several proposals of the regression upgrades in this thesis are also applicable to Aeschbacher et al.'s method.

Arad and Ben-Shahar's Method

In Arad and Ben-Shahar [5], we find K representative spectra using the K-SVD algorithm [3]:

$$\mathbf{D}^h = K\text{-SVD}^{\ell}(\mathbf{H}) = \left[\underline{\boldsymbol{h}}^1, \underline{\boldsymbol{h}}^2, \cdots, \underline{\boldsymbol{h}}^j, \cdots, \underline{\boldsymbol{h}}^K\right] \ , \tag{2.36}$$

where ℓ is the sparsity adopted by K-SVD, and the $\mathbf{H}$ matrix represents the training-set spectra. Then, following the RGB colour simulation process in Equation (2.4), we calculate the RGB counterpart of $\mathbf{D}^h$:

$$\mathbf{D}^c = \mathbf{R}^{\mathsf{T}}\mathbf{D}^h \ . \tag{2.37}$$

Here, like in RBFN, K representative RGBs are recorded in $\mathbf{D}^c$ (though in sparse coding usually a much larger K is selected), but unlike RBFN, these RGBs are not optimised from the RGB data but derived from the spectral representatives.

For the problem of spectral reconstruction, Arad and Ben-Shahar used the assumption that the neighbours in the RGB space are also neighbours in the spectral space (in the parlance of Timofte et al. [88], this assumption is called "neighbour embedding"). Following this assumption, given a query RGB $\underline{\boldsymbol{c}}_i$, we find a linear combination vector $\underline{\boldsymbol{w}}_i$ such that

$$\mathbf{D}^c\underline{\boldsymbol{w}}_i \approx \underline{\boldsymbol{c}}_i \ . \tag{2.38}$$

This $\underline{\boldsymbol{w}}_i$ vector is effectively estimated by the Orthogonal Matching Pursuit (OMP) algorithm [65]. Then, by applying this solved $\underline{\boldsymbol{w}}_i$ vector to the spectral dictionary $\mathbf{D}^h$, we achieve spectral reconstruction by

$$\mathbf{D}^h \underline{\boldsymbol{w}}_i \approx \underline{\boldsymbol{h}}_i \ , \tag{2.39}$$

where $\underline{\boldsymbol{h}}_i$ is the ground-truth spectrum corresponding to the input $\underline{\boldsymbol{c}}_i$.

Note that the components of $\underline{\boldsymbol{w}}_i$ are all positive and sum up to 1. And, the same sparsity (i.e., ℓ) that was used by K-SVD to train $\mathbf{D}^h$ is expected to be used by OMP when determining $\underline{\boldsymbol{w}}_i$. Given that usually $\ell \ll K$, most of the terms in $\underline{\boldsymbol{w}}_i$ will be zero (or very small). In other words, the spectral reconstruction from a given RGB camera response would only involve a small number of dictionary spectra.

Aeschbacher et al.'s "A+" Method

In Arad and Ben-Shahar's method, the idea of neighbour embedding operates at the *inter-representative* level. Indeed, the RGB representatives are the elements used by OMP to derive the linear combinations that estimate spectra (Equation (2.38) and (2.39)). In Aeschbacher et al. [1], a.k.a. the "A+" method, the idea of neighbour embedding is used on even more local data—the data in the *proximity* of each RGB representative.

In training, apart from $\mathbf{D}^c$, A+ also finds for each representative RGB in $\mathbf{D}^c$ the M closest data points in the training data set. Let us take the jth RGB representative in $\mathbf{D}^c$, i.e., $\underline{\boldsymbol{c}}^j$, as an example:

$$\mathbf{C}^j = \text{Prox}^M(\mathbf{C}, \underline{\boldsymbol{c}}^j) = [\underline{\boldsymbol{c}}^j_1, \underline{\boldsymbol{c}}^j_2, \cdots, \underline{\boldsymbol{c}}^j_i, \cdots, \underline{\boldsymbol{c}}^j_M] \ , \tag{2.40}$$

where $\mathbf{C}$ is the training RGBs, and $\mathbf{C}^j$ is the targeted data matrix that records the data points in $\mathbf{C}$ that are nearest to $\underline{\boldsymbol{c}}^j$. For both data matrices the columns are individual RGB data vectors. Here, additionally, the Prox^M function calculates and ranks the Euclidean distances of *normalised* vectors, i.e., both $\underline{\boldsymbol{c}}^j$ and the columns of $\mathbf{C}$ are normalised to unit-length when calculating the distances, and the superscript ^{M} indicates that top M closest data points are found. But, in the resulting $\mathbf{C}^j$ matrix, the un-normalised ground-truth vectors are recorded. We also record the ground-truth spectral counterparts of each column of $\mathbf{C}^j$ in the corresponding column of a spectral data matrix $\mathbf{H}^j$.

Then, the $\mathbf{C}^j$ and $\mathbf{H}^j$ local data matrices are used to replace the full $\mathbf{D}^c$ and $\mathbf{D}^h$ dictionaries used for linear combination in Equation (2.38) and (2.39). I.e., here we seek:

$$\mathbf{C}^j \underline{\boldsymbol{w}}_i \approx \underline{\boldsymbol{c}}_i \implies \mathbf{H}^j \underline{\boldsymbol{w}}_i \approx \underline{\boldsymbol{h}}_i \ , \tag{2.41}$$

where $\underline{\boldsymbol{c}}_i$ and $\underline{\boldsymbol{h}}_i$ are respectively the example query RGB and its ground-truth spectral counterpart, subject to:

$$\mathrm{Prox}^1(\mathbf{D}^c, \underline{\boldsymbol{c}}_i) = \underline{\boldsymbol{c}}^j \ . \tag{2.42}$$

Here, the Prox^1 function is similarly defined as in Equation (2.40), only that the searching data set is now the RGB dictionary $\mathbf{D}^c$, and only the closest one in $\mathbf{D}^c$ is found. Effectively, this step defines the locality of the map: the neighbour embedding assumption for the jth neighbourhood defined in Equation (2.41) is only adopted by the input RGBs in the same jth neighbourhood (whose closest representative in $\mathbf{D}^c$ is $\underline{\boldsymbol{c}}_j$).

In A+, the $\underline{\boldsymbol{w}}_i$ vector in Equation (2.41) is also solved in a different way compared to Arad and Ben-Shahar's method. Assuming the equivalence with the linear regression method (Equation (2.25)), we seek a local linear regression matrix $\mathbf{M}^j$ that satisfies:

$$\mathbf{M}^j \mathbf{C}^j \approx \mathbf{H}^j \ , \tag{2.43}$$

which, when solved in a least-squares sense (Equation (2.32)), suggests:

$$\mathbf{M}^j \underline{\boldsymbol{c}}_i = \mathbf{H}^j \left[{\mathbf{C}^j}^{\mathsf{T}} [\mathbf{C}^j {\mathbf{C}^j}^{\mathsf{T}} + \gamma \mathbf{I}]^{-1} \underline{\boldsymbol{c}}_i \right] \equiv \mathbf{H}^j \underline{\boldsymbol{w}}_i \approx \underline{\boldsymbol{h}}_i \ . \tag{2.44}$$

A key advantage of adopting the formulation in Equation (2.44) is that, evidently, all query RGBs in the same neighbourhood use the same regression matrix to derive their spectral estimates. That is, we can calculate all the regression matrices—$\mathbf{M}^j$ for $j = 1, 2, \cdots, K$—in training, leaving the inference step only to determine the neighbourhood labels of the query RGBs (Equation (2.42)).

In terms of SR performance, it is shown that Aeschbacher et al.'s A+ method performs on par with Galliani et al. [32]—an early implementation of the Deep Neural Network (DNN) based SR (the general idea of DNN will be introduced in the next section).

2.5 Deep Neural Networks

By virtue of the fast-increasing amount of data and the development of parallel processing technologies (i.e., the Graphics Processing Unit (GPU)), Neural Network approach has yielded a great leap of performance from its predecessors in a wide range of machine learning applications. Inspired by the structure of human brains, Neural Network is a learning approach based on the connections of *neurons*. These artificial neurons can be seen as some basic data-processing units, which are composed of three elements: *weights*, *bias* and *activation function*, and let us denote these three elements as $\underline{\boldsymbol{w}}$, b and $g()$, respectively.

Each neuron takes an input vector $\underline{\boldsymbol{a}}$—may be the actual input data, or the collection of outputs from the neurons in the previous *layer* in the structure—and map it to a scalar output a' following:

$$g(\underline{\boldsymbol{w}}^{\mathsf{T}}\underline{\boldsymbol{a}} + b) = a' \ . \tag{2.45}$$

For each neuron, $\underline{\boldsymbol{w}}$ and b are the trainable parameters. The activation function $g()$ is preset, with common choices such as ReLU (Rectified Linear Unit) and the sigmoid function [79].

There are *shallow* networks proposed for SR where only one neural layer in addition to the input and output layers (a.k.a. a *hidden layer*) is used. Examples include Sharma and Wang [77] and Ribés and Schmit [70]. Also the RBFN method [62] introduced in Section 2.4.1 is sometimes considered to be a shallow network. Similarly to the regression-based methods we introduced in previous sections, shallow networks are designed to find a one-to-one RGB-to-spectrum mapping. This limitation is overcome by the recent development of Deep Neural Networks (DNN), where much more than one hidden layers—i.e., a larger network "depth"—are used, which provide much better mapping abilities to solve even more complex fitting problems, such as, significantly, taking a whole *patch* of RGB image as input instead of the RGB at a pixel. Evidently, with RGB patches as inputs, the same RGB viewed in different contexts can potentially be distinguished by patch-based DNNs. This is a sensible approach to SR, since it addresses the metamerism phenomenon in real-world imaging conditions [29; 99], i.e., ground-truth spectra coming from different object surfaces can appear to have the same colour under a given lighting condition.

Compared to traditional machine learning methods (e.g., regression and sparse coding), DNN is able to learn more sophisticated feature representation from the data together with more complex mapping function in a distinct "end-to-end" manner. However, most of the Neural Networks do not have a closed-form global minimum solution—a local minimum is to be solved for using an iterative optimisation process. Also, the tuning for a functioning network architecture and the heuristic training setup for a particular application is still highly manual and laborious.

In SR, most of the DNN methods are based on the Convolutional Neural Network (CNN) or the Generative Adversarial Network (GAN) architectures [7; 80; 4]. In particular, CNN adopts the idea of replacing the weighting vector ($\underline{\boldsymbol{w}}$ in Equation (2.45)) with a convolutional operation in at least one layer of the network [37], and GAN consists of a generator network and a discriminator network (both networks are competing for a zero-sum game) [38], where typically a CNN is used as the discriminator and the generator runs a CNN in a reversed process (i.e., a "Deconvolutional" Neural Network). These methods provide top performances in recent benchmarks such as NTIRE 2018 and 2020 spectral reconstruction challenges [7; 8]. In this thesis, one of the top-performing methods in the NTIRE 2018 challenge [7], the CNN-based HSCNN-R [80], will be included in our benchmarks and dis-

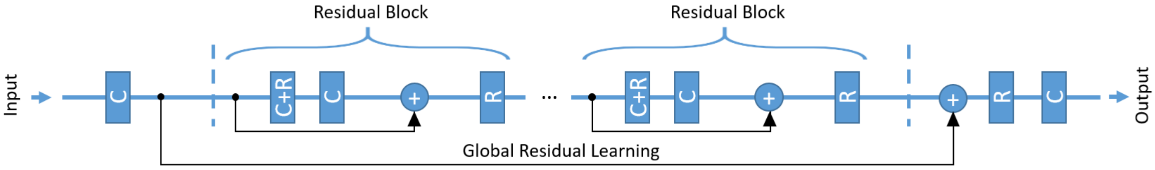


Figure 2.4: The HSCNN-R architecture. "C" means convolution with 3×3 filters (64 or 256 filters per layer, as suggested in [80]), and "R" refers to ReLU activation.

cussions. In NTIRE 2018, HSCNN-R was ranked first in one evaluation protocol and second in the other. The illustration of HSCNN-R is given in Figure 2.4.

The implementation of HSCNN-R is complicated, and so we point the readers to the original work [80] for more details. Of our concern, we note that the network takes $50 \times 50 \times 3$ (height $\times$ width $\times$ spectral dimension) RGB image patches as input and maps them to the corresponding $50 \times 50 \times 31$ hyperspectral image patches (the ground-truth hyperspectral images used for training have 31 spectral channels). Also, several variants were recommended for HSCNN-R in [80]. Specifically for the NTIRE 2018 challenge [7], they further adopted an ensemble method which means several different variants were trained whose outcomes were then averaged to provide the final recovery output. While the depth and per-layer neuron numbers of each variant vary, we calculated the number of their learnable parameters to be around 10^7 per variant. Evidently, HSCNN-R is orders-of-magnitudes more complex than the shallow-learned regressions, sparse coding and shallow networks (among which the A+ sparse coding method might have the highest number of parameters at around 10^5).

2.6 Summary

There is a huge body of literature dedicated to solving the RGB-to-hyperspectral image recovery (i.e., Spectral Reconstruction; SR). We have legacy methods which use explicit physical constraints to bound the recovery uncertainty, while recent proposals for SR are based on machine learning algorithms whose main target is to minimise the spectral recovery errors. Though the latter approach is generally believed to provide more accurate SR, these methods do not necessarily comply with the apparent physical constraints adopted by the legacy methods, e.g., whether the recoveries are realisable in the real world, or, for each recovery, whether we are actually searching for a solution among those could be the ground-truth (i.e., if the recovery is *physically plausible* to be the ground-truth or not). As one of the main objectives of this thesis, we are to reveal such physical compliance issues of the machine-learning-based algorithms, discuss on their practical implications, and finally develop ways to solve these issues.

We also see that machine learning methods of a wide range of complexities are used to solve the SR problem. Indeed, we have the regression-based methods where all learnable parameters are limited to a single linear transformation matrix, sparse-coding based methods incorporating an additional dictionary learning process (where the nearest neighbours search in these methods can be time consuming), and finally the DNN methods where thousands of learnable neurons are used, leading to orders of magnitudes more parameters used compared to regression and sparse coding. With a large discrepancy in complexity and the same single objective to minimise the overall spectral recovery error, we shall expect that the simpler regressions and sparse coding methods are no match for the leading DNN methods, right? As we embark on re-evaluating the relative performance of these machine learning methods on the same database, we are also going to propose regression and sparse coding upgrades so as to challenge the *assumed* dominance of DNN-based SR in the literature.

Chapter 3

Evaluating Spectral Reconstruction

In this section, we wish to establish the evaluation setup we are going to use throughout this thesis. To begin with, we conduct a baseline test which compares the considered SR methods using existing evaluation protocols. The effects of different spectral error metrics used and the adoption of a cross-validation procedure are explored.

Next we consider new parts to the evaluation methodology. First, we examine the role of camera spectral sensitivities in SR. Do different spectral sensitivities lead to different spectral recovery results? Also, we develop a methodology to consider the worst-case performance of SR algorithms. Part of the power of existing techniques comes from that they are trained on typical spectra found in typical images. Then, how do they fare when tested on images that have spectral and spatial statistics not represented in the training sets?

3.1 Baseline

3.1.1 Hyperspectral Dataset

The ICVL hyperspectral database [5], consisting of 200 images of both indoor and outdoor scenes, is used as the standard ground-truth database in this thesis. We use ICVL because it provides enough data to train a complex algorithm such as a DNN—a fact that was shown in the NTIRE 2018 SR challenge [7] where ICVL was used as the standard benchmark dataset. Other commonly used datasets in earlier works, such as the CAVE [101] and NUS [62] datasets, contain much fewer images while CAVE is further restricted to highly controlled indoor scenes.

Figure 3.1: Example scenes in the ICVL hyperspectral database [5]. The shown colour images were rendered only for the displaying purpose.

In ICVL, most images have the spatial dimension of 1392 × 1300 while some scenes are 1392 × 1089. There are 31 spectral channels, representing the spectral measurement from 400 to 700 nanometers (nm) with 10-nm intervals. And, the data is stored in 12 bits, i.e., the pixel values range from 0 to 4095. Several example images included in the ICVL dataset are shown in Figure 3.1.

3.1.2 RGB Image Simulation

For training and evaluation, we need matching ground-truth RGB and hyperspectral data. In line with other research in this area, our experiments are based on ground-truth hyperspectral images where the RGB counterparts are generated by numerical integration (see Equation (2.4) in Section 2.1.3). More particularly, we adopt the NITRE challenges' "clean track" methodology [7; 8; 9]. Additionally, in clean track, there is the constraint that the RGB images are formed by numerically integrating hyperspectral data using the CIE 1964 colour matching functions [24].

3.1.3 Cross Validation

The evaluation cycle of a learning algorithm consists of three phases: training, validation and testing. In training, we update the parameters of the model so that it fits the training data well. While the trained model could face the overfitting problem (see, e.g., Section 2.3.3), we need to tune the model's construct (normally the model's ability to fit the training data) to ensure its generalisibility to another set of data outside the training set, i.e., the validation set. This *validation* process can be different depending on the SR approaches. For

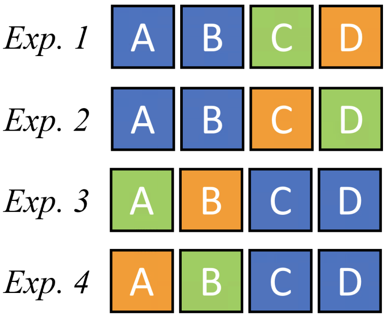


Figure 3.2: Our cross validation setup. Each coloured squared block represents equal amount of randomly allocated data. The blue, green and orange patches represent the data for training, validation and testing, respectively. “Exp.” is short for “Experiment”.

regression-based models, we determine the regularisation parameter introduced in Section 2.3.3 (Equation (2.31) and (2.32)), and for a DNN, the performance on the validation set tells us when to stop its iterative training process. Finally, the trained and validated model is used on the testing data to deliver the final performance evaluation.

We adopt a 50%-25%-25% random data partition for training, validation and testing. Since there are 200 images in the ICVL database, we randomly separate all images into 4 groups of 50 scenes, and then use 2 groups for training, 1 group for validation, and 1 group for testing. We call this the Single Validation (SV) setting. The potential problem of SV is that the random partition might be *unfair* splits, i.e., the performance evaluation might change if we switch around the *purpose* (training, validation, or testing) of each image group.

To address this issue, we create a Cross Validation (CV) setup shown in Figure 3.2. Here, each squared block represents a group of 50 scenes, respectively labeled as A, B, C and D. The colour of each block indicates their purpose, with blue blocks indicate training, green for validation, and orange for testing. In this way we run 4 different SV experiments, and we get recovery results for the 4 test sets which, when taken together, comprise exactly the whole image set. We will calculate performance measures on the SR recoveries for the 4 test sets. We then average these measures to arrive at the recovery evaluation for the whole image set.

Clearly, a CV setup is more time consuming than an SV. This problem is more pronounced for the DNN-based methods, for their much longer training time. In this thesis, where possible, we use the better CV setup, while SV is still reported sometimes for quicker evaluations.

3.1.4 Evaluation Metrics

The following 4 metrics are commonly used in the literature: Mean Relative Absolute Error (MRAE), Root-Mean-Square Error (RMSE), Angular Error (AngE), and Peak Signal-to-Noise Ratio (PSNR). Denoting $\Psi(\underline{\boldsymbol{c}}_i)$ as the recovered spectrum from the RGB $\underline{\boldsymbol{c}}_i$, and $\underline{\boldsymbol{h}}_i$ as the target ground-truth spectrum, these metrics are defined as follows:

- Mean Relative Absolute Error:

$$\text{MRAE (\%)} = 100 \times \frac{1}{n}\left|\left|\frac{\Psi(\underline{\boldsymbol{c}}_i) - \underline{\boldsymbol{h}}_i}{\underline{\boldsymbol{h}}_i}\right|\right|_1 , \tag{3.1}$$

  where n is the number of spectral channels (in our case $n = 31$), the division is element-wise and the ℓ_1 norm is calculated. Essentially, this MRAE metric measures the averaged percentage absolute deviation over all spectral channels. Note that this metric becomes unstable when the ground-truth values of any spectral channels approach zero (which could be prevented if we calculate the error relative to the norm of the ground-truth spectra instead), but we will continue to use this metric as it is the standard for ranking and evaluating SR algorithms in the recent benchmark [7; 8; 9].

- Root-Mean-Square Error:

$$\text{RMSE} = \sqrt{\frac{1}{n}||\Psi(\underline{\boldsymbol{c}}_i) - \underline{\boldsymbol{h}}_i||_2^2} . \tag{3.2}$$

Unlike MRAE, RMSE is *scale dependent*, that is the overall brightness levels of the compared spectra and/or the data encoding bit-depth will reflect on the scale of RMSE.

- Angular Error:

$$\text{AngE} = \cos^{-1}\left(\frac{\Psi(\underline{\boldsymbol{c}}_i)}{||\Psi(\underline{\boldsymbol{c}}_i)||} \cdot \frac{\underline{\boldsymbol{h}}_i}{||\underline{\boldsymbol{h}}_i||}\right) \,, \tag{3.3}$$

where the angle between the compared spectra is calculated. Another commonly used metric equivalent to AngE is the Goodness-of-Fit Coefficient (GFC) [72], which is the cosine of AngE (i.e., only calculating the inner product of normalised vectors). Uniquely, AngE does not measure the difference in brightness scale between spectra: only the *shapes* of the spectra are compared.

- Peak Signal-to-Noise Ratio:

$$\text{PSNR (dB)} = 10 \times \log_{10}\left(\frac{v_{\max}^2}{\frac{1}{n\times N}\sum_{i\in I}||\Psi(\underline{\boldsymbol{c}}_i) - \underline{\boldsymbol{h}}_i||_2^2}\right) \,, \tag{3.4}$$

where $v_{\max} = 2^{12} - 1 = 4095$ is the maximum possible value for 12-bit images, n is the number of spectral channels, and N is the number of pixels in image I. As shown here, PSNR is usually defined at the image level, as opposed to other three metrics where errors are calculated per pixel.

3.1.5 Respective SR Model Settings

Table 3.1: Considered spectral reconstruction methods.

Abbreviation	**SR method**	**Type**
LR	Linear Regression [42]	Regression
PR	Polynomial Regression [25]	Regression
RBFN	Radial-Basis-Function Network [62]	Clustering + Regression
A+	A+ Sparse Coding [1]	Clustering + Regression
HSCNN-R	HSCNN-R Network [80]	Deep Neural Network

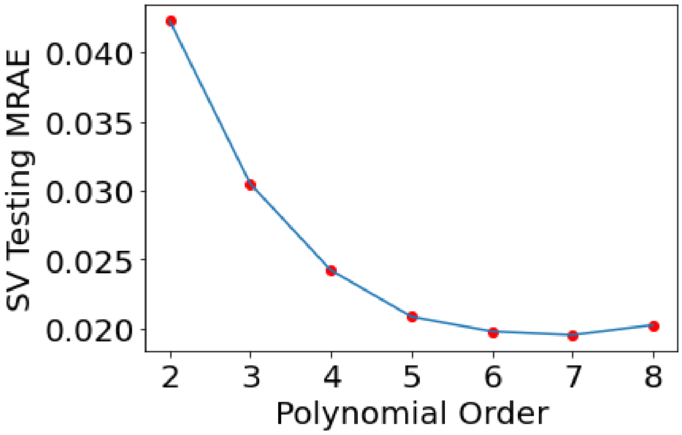


Figure 3.3: Single-validated (SV) regularised polynomial regression performances with respect to the polynomial expansion orders.

The SR algorithms that are considered throughout this thesis are listed in Table 3.1. Since most of these methods (all except the LR method) have several possible and/or suggested variants, here, we are to specify the variants we will use in this thesis.

Polynomial Regression

As mentioned in Section 2.3.1, a higher-order polynomial regression provides a more complex mapping function at the risk of overfitting. A primary test on the appropriate order to use for the *regularised* (Section 2.3.3) polynomial regression is shown in Figure 3.3. Here, the single-validated (SV) MRAE performance is plotted with respect to the polynomial order used. Clearly, the lowest errors (best performances) occur when the polynomial order is around 6 or 7. As a result, in this thesis we use the 6th-order regularised polynomial regression.

Radial-Basis-Function Network

The original work [62] suggests using 45 to 50 centres for the radial basis functions (see Section 2.4.1 for more details). Hence, in this thesis we use 45 centres for the RBFN method.

A+ Sparse Coding

The two main factors that decide the A+ model's performance, as introduced in Section 2.4.2, are the number of representatives, K, and the number of nearest neighbours considered around each representative, M. Both factors are also relative to the amount data used for training. In the original work [1], 3000 pixels per training image (that is 300,000 data points in total for 100 training images) were randomly selected for the K-SVD training, with $K = 1024$. Then, the nearest neighbours are found in a larger set of training data: 30000 pixels per training image (3,000,000 data points), with $M = 8192$. We will continue to use these setups in our experiments.

Another two factors to consider are the sparsity setting (ℓ in Equation (2.36)) used when running the K-SVD algorithm and the regularisation parameter used for all local linear regressions. We retain the same sparsity setting used in [1], which is $\ell = 8$. However, for regularisation we optimise individual regularisation parameters for each local linear regression separately, following the regularisation approach introduced in Section 2.3.3.

HSCNN-R Network

According to [80], we can adjust the depth of HSCNN-R. We adopt one of the recommendations which uses a 20-layer depth and 256 filters per convolutional layer. In training and validation, the ending *epoch* (i.e., the number of rounds that the whole training set is used to update the DNN's model parameters) was set empirically at around 300 to 350.

3.1.6 Testing Results

We begin by reporting, in Table 3.2, the training times required for the various methods along with the number of parameters inherent to each method. We calculate the training time as the average time used for each single-validation experiment (Section 3.1.3), and the reconstruction time as the average time used to reconstruct each tested image. We also present the number of parameters used in each algorithm, according to the algo-

Table 3.2: The number of parameters count and the training and reconstruction time measurements.

Model	Number of Parameters	Training Time (single validation)	Reconstruction Time (per image)
LR	93	6.7 min	0.031 s
PR	2573	15.1 min	6.0 s
RBFN	1426	1.0 h	3.1 s
A+	9.5×10^4	26.9 min	13.7 s
HSCNN-R	1.1×10^7	35.5 h (with GPU)	13.0 s (with GPU)

rithm variants specified in Section 3.1.5. Of course, any timings are related to the hardware used in the experiments. Our hardware specification includes Intel® Core^{TM} i7-9700 CPU and NVIDIA® GeForce® RTX 2080 SUPER^{TM} GPU. Note that the GPU is only used for the DNN-based HSCNN-R, for both training and reconstruction. All other regression algorithms use solely the CPU.

It is evident that HSCNN-R is more complicated than the regressions as evidenced by the relatively high numbers of parameters and the training and reconstruction timings. Indeed, the HSCNN-R network—even with a GPU boost—still takes days to train, whereas for regressions the training time is mostly lower than an hour (for the clustering-based methods, A+ and RBFN, the main portion of the training time is used on training the dictionaries).

Of course, while training can be carried out offline, the reconstruction of spectra is a real-time task. Thus it is the reconstruction time that speaks most strongly to whether a method is likely to be practically useful. Where, we see that HSCNN-R, while GPU is deployed for its reconstruction, still spends more time to reconstruct an image than most of the regression-based methods. Note that for all regressions except LR, the reconstruction times already appear objectively long. This is because, unlike LR only involves a matrix multiplication, other regressions require either non-linear transformations, e.g., in PR and RBFN, or nearest neighbour search, i.e., in A+. Improving the execution speed of these additional processing steps will be key to making these algorithms more practical.

Table 3.3: Cross-validated mean per-image-mean errors of the SR methods. Best results are in bold and underlined.

	Mean Per-Image-Mean		
Method	**MRAE (%)**	**RMSE**	**AngE (deg)**
LR	6.24	33.26	3.79
A+	3.87	23.97	2.39
RBFN	2.06	18.30	1.49
PR	1.95	17.05	1.46
HSCNN-R	**1.73**	**16.33**	**1.34**

Table 3.4: Cross-validated mean per-image-99-percentile errors of the SR methods. Best results are in bold and underlined.

	Mean Per-Image-99-Percentile		
Method	**MRAE (%)**	**RMSE**	**AngE (deg)**
LR	16.95	99.76	9.45
A+	15.26	94.18	8.75
RBFN	7.89	81.88	4.97
PR	7.10	**75.56**	4.87
HSCNN-R	**6.53**	77.21	**4.69**

Table 3.5: Cross-validated mean PSNR errors of the SR methods. Best results are in bold and underlined.

Method	Mean PSNR (dB)
LR	41.35
A+	43.52
RBFN	45.57
PR	46.09
HSCNN-R	**46.30**

Next, we present the SR performance of the methods. Because the hyperspectral "images" are reconstructed, we are interested in the "per-image" statistics, including the average of the mean and the average of the worst-case errors in individual images (except for the PSNR metric which is already a per-image measurement). We show the mean per-image-mean and 99-percentile statistics in Table 3.3 and 3.4, respectively. The PSNR statistics are presented separately in Table 3.5. Note that for MRAE, RMSE and AngE, the smaller the number, the better the performance, while for PSNR, larger numbers mean better performance.

In almost all cases, the performance data shown in these tables are in the order of increasing performance. That is the DNN-based HSCNN-R is the best overall. The exception is RMSE where the polynomial regression method performs better in the per-image 99 percentiles.

It is interesting to consider why the PR method might perform better in regard to the RMSE performance criterion. We note that regressions like PR are optimised for a least-squares criterion (Section 2.3.2), i.e., minimising squared RMSE. In contrast, HSCNN-R minimises MRAE [80]. This result draws attention to the fact that the metric used in training is important in algorithm performance. We will explore this issue in more detail in Chapter 4 where we show how we can elevate the performance of all regression methods by modifying the regression error to match the MRAE used in training the DNNs.

3.2 Evaluation on Different Cameras

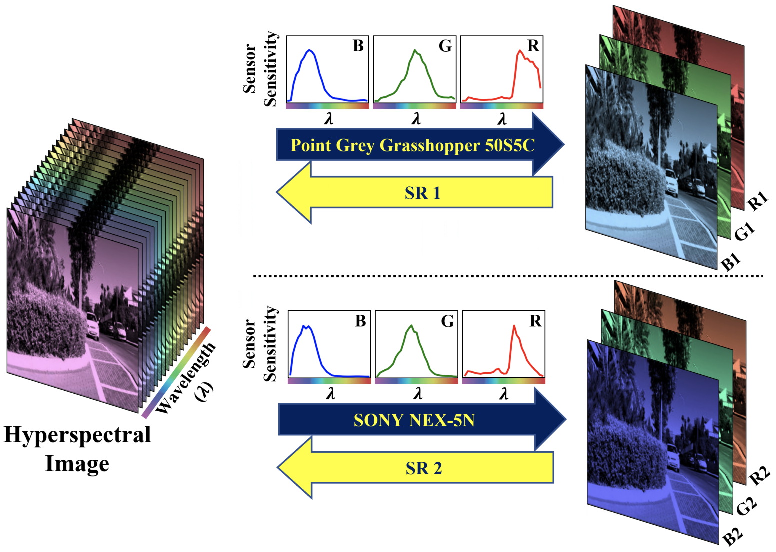


Figure 3.4: The RGB imaging outcome depends on the used camera model, and so SR, as the reverse process, should also be dependent of the camera model (i.e., SR 1 $\neq$ SR 2).

Different camera manufacturers and/or models use different sets of colour sensors that have different spectral sensitivity characteristics (Section 2.1.1). As illustrated in Figure 3.4, the raw RGB camera responses of the same object (under the same light) might be different from one camera model to another. Here, we evaluate the extent to which the SR algorithms' performance depends on the camera's spectral sensitivities.

In the prior art, Arad and Ben-Shahar [6] demonstrated that there exists a significant difference in recovery accuracy for clustering-based SR algorithms when different cameras are used, and Kaya et al. [47] addressed this issue by developing a DNN-based method where RGB images from different cameras can all be admitted as input. Recently, Fu et al. [30] further proposed a CNN-based model that jointly selects the best camera sensitivities and recovers spectra. While some aspects of the problem of switching cameras in SR have been addressed in these works, we realised that there was not a comprehensive investigation of the SR performance *as a function of both camera and SR algorithm.*

In this section, we aim to show empirically how switching cameras for SR can affect the SR performance and the rankings of the algorithms.

3.2.1 Experiments

Table 3.6: Considered camera models for simulating the RGB images from the hyperspectral images. The number behind each "Cam" in aliases corresponds to the order index in the original RIT database of camera sensitivities [45].

Alias	**Camera name**
CMF	CIE 1964 Color Matching Functions
Cam 0	Canon 1D MarkIII
Cam 9	Hasselblad H2
Cam 10	Nikon D3X
Cam 20	Nokia N900
Cam 21	Olympus E-PL2
Cam 22	Pentax K-5
Cam 24	Point Grey Grasshopper 50S5C
Cam 26	Phase One
Cam 27	Sony NEX-5N

The RIT camera sensitivity database [45] provides spectral sensitivity functions of 28 commercially available RGB cameras. To maximise the variety of our camera selection, from this database we select 9 cameras which are all from different brands. Additionally, we include the CIE 1964 colour matching functions [71] into consideration, not least because they were used in the NTIRE challenges [7; 8; 9] and throughout this thesis. These 10 selected cameras and their aliases used in this section are listed in Table 3.6.

Based on these 10 sets of camera's spectral sensitivity functions, we generate 10 different RGB image sets from the 200 ICVL hyperspectral images [5] following the colour image formation formula (Equation (2.4)). Then, for all the methods listed in Table 3.1, we evaluate their SR performances while these 10 sets of RGB images are in turn used as the input dataset. Also, the MRAE error metric and the single validation (SV) methodology (Section 3.1.3) are adopted here.

3.2.2 Results

Table 3.7: The single-validated mean per-image-mean MRAE results for the switching camera experiment.

	Mean Per-Image-Mean MRAE (%)									
Method	**CMF**	**Cam 0**	**Cam 9**	**Cam 10**	**Cam 20**	**Cam 21**	**Cam 22**	**Cam 24**	**Cam 26**	**Cam 27**
LR	6.39	6.31	6.13	6.23	6.09	6.16	6.21	5.96	6.52	6.29
A+	3.81	4.01	3.84	3.98	3.64	3.92	3.89	4.50	3.95	3.75
RBFN	2.10	1.89	1.86	1.90	1.93	1.91	1.87	1.80	2.09	1.94
PR	1.98	1.86	1.80	1.86	1.80	1.83	1.84	1.75	1.98	1.88
HSCNN-R	1.76	1.67	1.65	1.71	1.72	1.69	1.69	1.63	1.77	1.69

Table 3.8: The single-validated mean per-image-99-percentile MRAE results for the switching camera experiment.

	Mean Per-Image-99-Percentile MRAE (%)									
Method	**CMF**	**Cam 0**	**Cam 9**	**Cam 10**	**Cam 20**	**Cam 21**	**Cam 22**	**Cam 24**	**Cam 26**	**Cam 27**
LR	17.26	14.96	14.30	14.69	15.06	14.54	14.68	13.43	16.24	15.37
A+	15.51	13.57	13.30	13.61	14.10	13.93	13.52	12.26	14.57	13.94
RBFN	8.77	7.75	7.49	7.57	7.65	7.92	7.29	7.39	8.71	7.48
PR	7.89	6.89	6.70	6.91	6.78	6.81	6.76	6.81	7.54	6.99
HSCNN-R	7.40	6.62	6.24	6.49	6.46	6.36	6.38	5.69	7.33	6.69

In Table 3.7 and Table 3.8, respectively, we present the mean and worst-case performances of the methods. Additionally, for the mean results, we conduct a Paired Student's t-Test [81] between the best camera's and worst camera's results, which is shown in Table 3.9.

The Student's t-Test examines statistically whether two sampled data distributions (here, the distributions of errors) differ in their mean value with significance. And, the *paired*

Table 3.9: Student t-test results between the mean performances (Table 3.7) of the worst and best cameras used for each SR method.

Method	Worst Cam	Best Cam	t-score	p-value
LR	Cam 26	Cam 24	4.33	$< 10^{-4}$
A+	Cam 24	Cam 20	3.99	$< 10^{-3}$
RBFN	CMF	Cam 24	4.78	$< 10^{-5}$
PR	Cam 26	Cam 24	5.47	$< 10^{-6}$
HSCNN-R	Cam 26	Cam 24	3.88	$< 10^{-3}$

version of the test signifies that the two samples are *dependent*—in our case, the compared test results are recovery errors of the *same* set of test scenes. Also, the *one-sided* hypothesis is used, where only the significance of the best camera being better—not worse—is tested. We present both the raw t-scores and the corresponding p-values in Table 3.9. As shown, the performance of the best camera is always statistically better than the worst for all 5 algorithms tested (i.e., $p < 0.05$, or equivalently, at $> 95\%$ statistical significance).

For one scene in the ICVL dataset, we present the pixel-wise MRAE error maps representing the performance of the worst and the best cameras for each SR method in Figure 3.5.

3.2.3 Discussion

The statistics in Table 3.7, 3.8 and 3.9 show several important results. First of all, let us fix each SR method individually while comparing the performance when different cameras are used (i.e., comparing the numbers in the tables horizontally). Clearly, for all methods, both the mean and 99-percentile performances vary when different cameras are used. And, the range of variation (between worst and best cases) for mean results is around 9% to 24%, and for 99 percentiles it is in the order of 18% to 30%, depending on the algorithms.

In general, SR methods with lower mean errors have lower 99-percentile errors. However, the respective camera rankings based on either mean or worst-case performances do not always match, i.e., cameras that return better mean performance might perform worse in the worst case. We can visually observe this discrepancy in Figure 3.5. Clearly, while the best cameras lower the general level of errors, some parts of the scene appear to get worse,

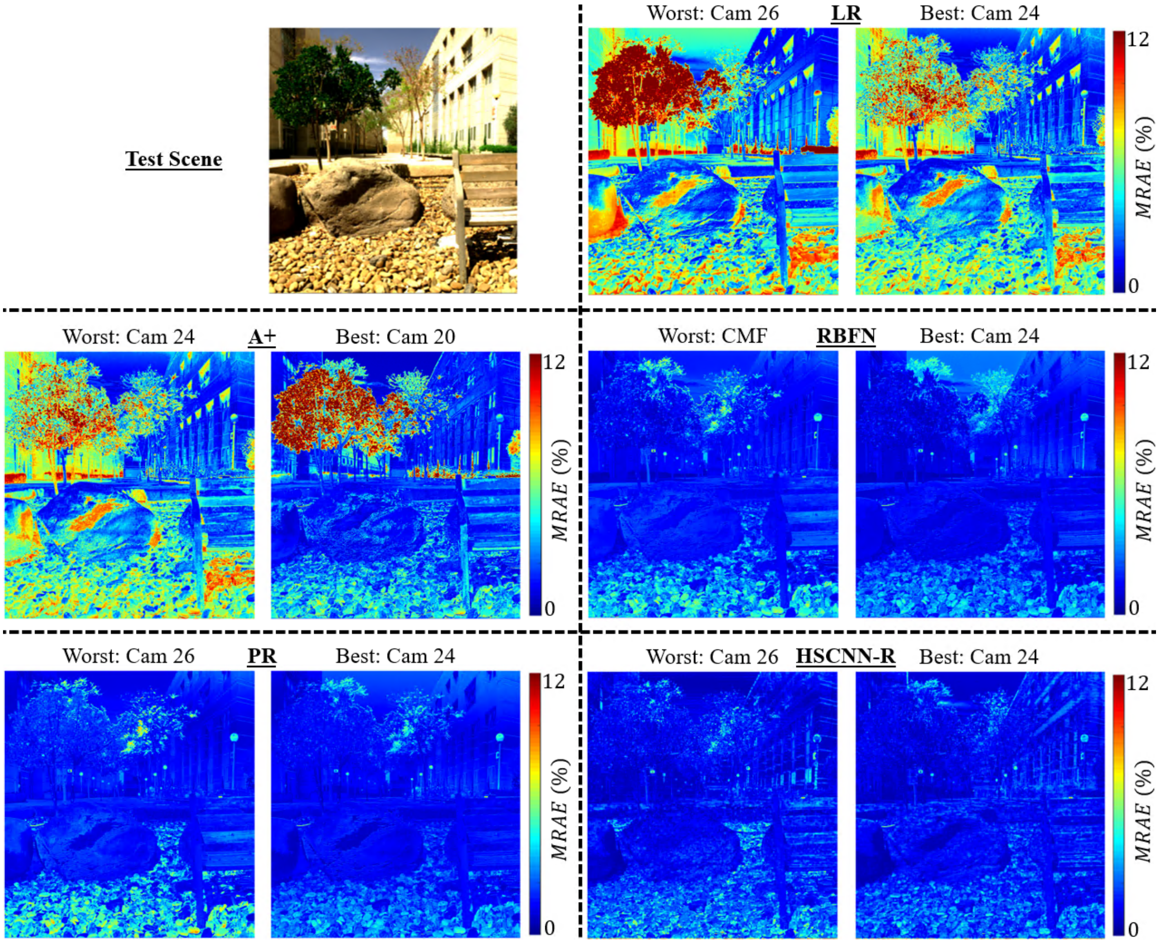


Figure 3.5: The worst camera vs. best camera comparison for each SR method on one example scene, in terms of the MRAE error heat map.

e.g., the "tree" part in the top-left corner for A+, and the "sky" part in the top portion for PR and RBFN.

In Table 3.9, we also observe that, in terms of mean performance, the best camera for one method is not necessarily the best for other methods. For instance, the best camera for LR, RBFN, PR and HSCNN-R—Cam 24—is in turn the worst camera for A+. This is a curious result and indicates a potentially rather strong dependence between camera and algorithm.

Next, let us consider the rankings of SR methods while fixing each camera used (i.e., comparing the numbers vertically). In terms of the mean performance, we see the rankings do not change when switching the cameras. This result provides an empirical basis that we could choose from a range of real camera spectral sensitivities to benchmark the SR methods with a consistent ranking.

Finally, let us view the results in the context of changing both the cameras and algorithms. In SR—as in many areas of computer vision—it is evident that we need to be careful in experiments and "compare like with like". Suppose that a first team of researchers train and test the PR method using Cam 24 (Point Grey Grasshopper 50S5C) while another team trains and tests HSCNN-R using Cam 26 (Phase One). If we directly compare the two performances, we might reach the conclusion that PR performs slightly better than HSCNN-R in mean performance, and in terms of 99 percentiles the performance advantage is significant. Yet, if we train and test the two methods on the same camera, then either Cam 24 or Cam 26 will suggest that HSCNN-R is better.

Another aspect to look at this comparison is that, as we are pursuing the advance of SR mapping function (in most case, make it more complex), it is possible that switching the used RGB camera can reach better performance—in this case, we are talking about switching the camera used for training PR from Cam 26 to Cam 24, which can reach better performance than continuing using Cam 26 and switching to use the much more complex HSCNN-R.

3.3 The Worst-Case Radiance-Mondrian-World Assumption

In most of the recent works, and also so far in this thesis, SR methods are trained on hyperspectral image database of real-world scenes. The testing images—the images that are used to evaluate the SR methods—usually also come from the same database. Naturally, we can expect those testing images to look broadly similar to the training images. For particular applications where all scenes look alike, this experimental setup might be adequate. In this section we are interested in the SR algorithms' performance on *unseen and unexpected* images.

We are going to propose a worst-case image set that bounds how well any SR algorithm can perform. Our goal is to design an inherently unseen and unexpected imaging condition that is theoretically realisable in the real world but provides as few clues as possible for learning. More specifically, we seek to ensure the image content is generated randomly, with unpredictable spatial patterns and spectral radiances. Then, using this newly defined and generated testing image set, we are to examine if more complex SR methods still perform better than the simpler ones.

3.3.1 Mondrian-World Patterns

For the random spatial patterns, we take inspiration from the Mondrian-World (MW) patterns [50] which are commonly used in colour research. A common definition of an MW image is a patchwork of overlapping rectangles with randomly generated locations and sizes. Figure 3.6 shows an example MW image.

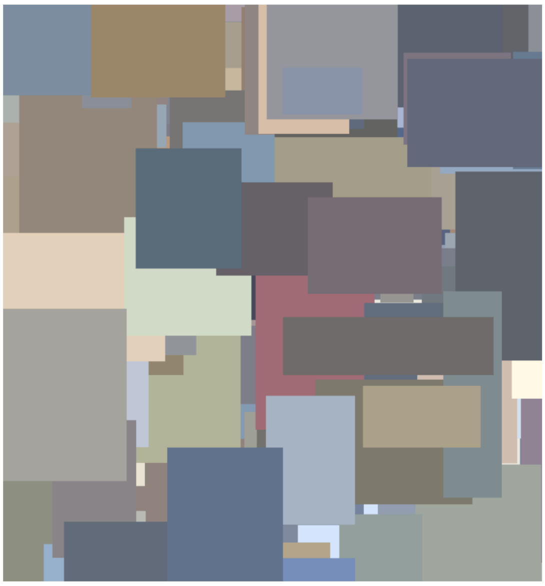


Figure 3.6: An example Mondrian-World (MW) scene. Each colour indicates a unique reflectance or radiance.

Clearly, MW images are not like everyday images. This is entirely the point. They are meant to be very simple stimuli devoid of any cues that the vision system might exploit.

There is no shape, no shading, no depth and no texture. And therefore, these images are a particular challenge for any learning-based algorithms (including those for spectral reconstruction) that attempt to exploit structures in images.

To automatically and randomly generate MW images, we take the liberty to formulate a simulation process:

1. Start with a blank array of height H and width W, referred to as the *canvas*.
2. Decide a rectangle height by drawing a number from the normal distribution $\mathcal{N}\big(\mu = \frac{H}{5}, \sigma = \frac{H}{15}\big)$ and a rectangle width drawn from $\mathcal{N}\big(\mu = \frac{W}{5}, \sigma = \frac{W}{15}\big)$. Both numbers are rounded up to integers.
3. Draw another 2 integers from uniform random distributions ranging [1, H] and [1, W], respectively, as the coordinates of the top-left corner of this rectangle.
4. Insert this rectangle to the *canvas*, overlaying on whichever patches that have already been on the *canvas*.
5. Repeat Step 2 to 4 until all pixels on the *canvas* are filled.

Following this process, at each random generation process, we decide the pattern of one particular MW image. Then, the next step is to decide what exactly should be filled in each of these patch regions.

Note that in this generation process we adopted a fixed probability distribution for the rectangle size. This was for generating patterns visually comparable to the Mondrian pattern when it was first proposed [50]. Of course, if the SR algorithms are trained on those images, the fixed probability distribution might be learned. Hence, for further research, one could seek to also relax the restriction on the probability distribution.

We also want to point out that training/testing with MW patterns only affects the capability of the patch-based methods, i.e., the DNNs. Indeed, for pixel-based algorithms, the scene content is never used (so removing it does not harm pixel-based algorithms). In the next sections, we will propose the other half of our design for the worst-case imaging scenario which will also impact the pixel-based algorithms.

3.3.2 The "Radiance" Mondrian World

In the literature of colour applications where the MW patterns are used, each patch region is expected to be uniformly filled by one *reflectance*—which means, each area is made of one material, or painted by the same paint, or printed by the same ink, etc. However, most MW images are also assumed to be illuminated by a *single* light source. As we are dedicated to finding a real-world worst-case scenario, we seek to further remove this prior knowledge (that different patches are illuminated by the same light), in order to create a condition that is even more challenging than MW.

Therefore, we propose the Radiance-Mondrian-World (RMW) imaging condition. In RMW, each patch of the MW pattern is filled with a real-world *radiance*. In terms of a potentially practical fabrication process, every patch area of the MW pattern is not just made of an arbitrary material, but **also** illuminated by an arbitrary light source. As we propose this new RMW condition, here, we are also the first to admit that such a patch-wise variation in illumination does not readily occur. However, we could certainly make an RMW image in the lab. Further, in the field of multi-illuminant color constancy, scenes with considerable spatial variation in illuminating light (that are plausible scenes) are of interest. For example, in the Figure 2 of Hao et al. [40] they consider scenes lit by 4 different spectra where, per pixel, the illumination is a convex combination of these 4 light spectra. Given their experimental setup it is easy to envisage scenes with even more light spectra thus taking us toward the RMW—if Hao et al. [40] is proposing 4 lights as a plausible scenario for multi-illuminant colour constancy, then clearly another research will propose 5, then 6, and so on. However, honestly, the feasibility of our RMW assumption is not a huge concern for us. Rather, we use it as a mean to theoretically bound a worst-case performance.

Here, we use a *simulation* process to generate the RMW images. More specifically, given a randomly generated Mondrian-World pattern, we are to fill each patch region with a randomly selected radiance spectrum from a pool of **eligible radiances**. To link RMW to the real world, for example, we might want to assign all spectra that appear in the ICVL hyperspectral image database [5] to be the eligible radiance set. Nevertheless, there is a bias of relative abundance of particular radiance spectra in regular scenes—e.g., the spectra of the sky, the trees, the ground, and the like, those commonly appeared image contents. Hence, we are also seeking to further exclude this bias when assigning the eligible radiance set.

3.3.3 The Convex Set of Natural Radiances

Following Finlayson and Morovic [29], we restrict the eligible radiances to be in the *convex closure* of all radiances found in the ICVL dataset. This convex closure idea is important since it integrates an otherwise discrete set of image sample points. Moreover, a patchwork of spectra of relative proportions (that sum up to one) appears like the same convex sum of spectra when viewed from a far enough distance away (so that all the different surfaces map to the same pixel). This sub-pixel integration concept is illustrated in Figure 3.7. On the left of the figure we have an RMW image. As we zoom in to the marked pixel, we see that what looks like a beige colour is actually composed of many different sub-pixel colours/radiances as shown on the right (i.e., a smaller, sub-pixel-sized RMW image). Lastly, as a continuous convex region we are able to randomly and uniformly sample the closure area to ensure no one natural spectrum is more likely to be selected than another.

Building a convex set in high dimension can be difficult. Indeed, in our case, the dimension of the spectra (i.e., the number of spectral channels in those ICVL hyperspectral images) is 31, which is way too high for, e.g., the Qhull [11] implementation. As a result, we are to find a lower-dimensional linear model using the PCA technique [69] to represent the spectral data (the idea of linear model was introduced in Section 2.2.1). Here, we use an

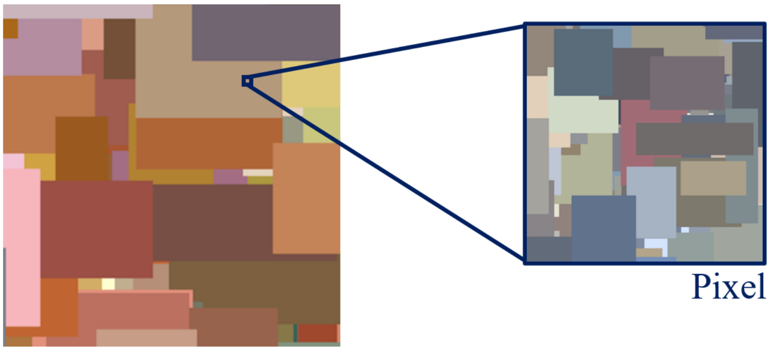


Figure 3.7: A radiance spectrum in the convex closure of natural radiances can be viewed as a sub-pixel mixture of the natural radiances, which makes such a spectrum also a realisable natural radiance spectrum.

8-dimensional linear model:

$$\underline{\boldsymbol{h}}_i \approx \mathbf{B}^h \underline{\boldsymbol{\alpha}}_i^h \ ; \ \ \forall\ i, \tag{3.5}$$

where $\mathbf{B}^h$ is a 31×8 matrix of basis vectors, and $\underline{\boldsymbol{\alpha}}_i^h$ is the unique 8-dimensional characteristic vector that defines the best fit to $\underline{\boldsymbol{h}}_i$, i.e., the ith radiance spectrum in the database. We choose to use an 8-dimensional representation because, based on some prior works, most real-world spectra can be described by a 5- to 8-dimensional linear model representation [64; 41; 72; 51; 59].

With respect to this linear model representation, we convert all spectra in the ICVL dataset to their 8-dimensional characteristic vectors. Then, we calculate the 8-dimensional convex hull (i.e., the parameters of the surrounding surfaces of the convex set that define the convex region) of these characteristic vectors. We denote this convex region as $\mathcal{C}$.

Next, with the $\mathcal{C}$ region calculated, we wish to randomly and uniformly sample this region. Our accept-reject sampling methodology is illustrated in Figure 3.8 (here, a 3-dimensional analogy is presented). In the figure, the green region represents the 8-dimensional convex set $\mathcal{C}$. Then, on the outside of $\mathcal{C}$ we find its *bounding box*, denoted as $\mathcal{B}$ (red region), which is the smallest cube that contains $\mathcal{C}$. Mathematically, $\mathcal{B}$ is delimited by the min and max value per dimension of $\mathcal{C}$. With respect to $\mathcal{B}$, it is very easy to sample data randomly and uniformly—we simply need to randomly and uniformly select a number within the bound of $\mathcal{B}$ in each dimension. Then, for each point sampled within the bounding box we can check whether this sample also falls inside $\mathcal{C}$: if it is inside $\mathcal{C}$ we *accept* the sample (to be converted back to spectral dimension by Equation (3.5) and used to fill the RMW patch), otherwise we *reject* it and re-select another sample from $\mathcal{B}$.

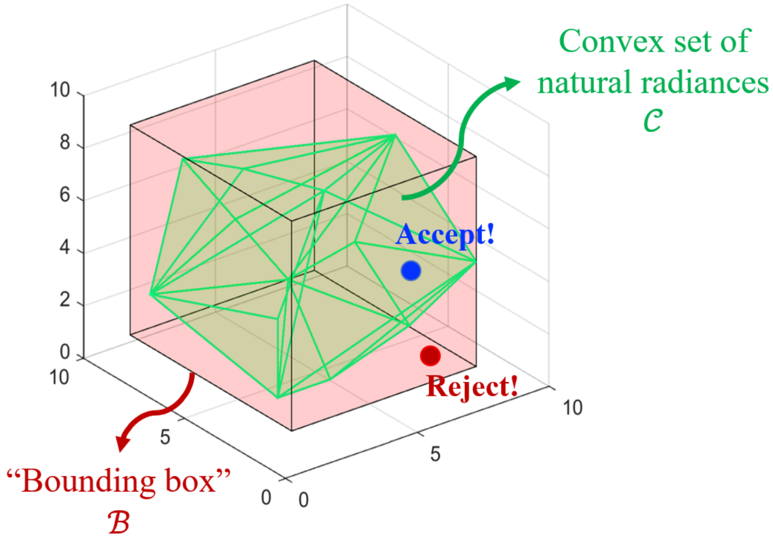


Figure 3.8: A 3-dimensional illustration of a convex set $\mathcal{C}$, its bounding box $\mathcal{B}$, and the accept-reject sampling strategy.

Combining the MW spatial pattern generation strategy introduced in Section 3.3.1 and the radiance selection methodology here (selecting new random radiances region-by-region in the MW pattern), an RMW image can be simulated.

3.3.4 Experiments and Results

With the definition and generation of the RMW images, we conduct 2 experiments. In the first experiment, all SR methods are trained and validated *as normal* using the original ICVL data, but then we test them on 50 random RMW images we generated. Here, we use the single validation process introduced in Section 3.1.3 while swapping the testing image set with our RMW set. Also, the CIE 1964 colour matching functions [24] are used as the camera's spectral sensitivities to generate the ground-truth input RGB images. Hereafter we refer this experiment to as the **Original/RMW** experiment. In effect, **Original/RMW** investigates how well the SR methods that are trained on the original hyperspectral images can *generalise* to the RMW images.

In the second experiment, we retrain the SR methods on the additionally generated 150 RMW images (100 images for training and 50 images for validation), while testing on the same RMW testing set used in the first experiment. This experiment is referred to as the **RMW/RMW** experiment. In this case, we are studying these methods' ability to actively learn to recover RMW images.

The mean and standard deviation (Std) of the per-image-mean MRAE and AngE results of the single-validated baseline testing (Section 3.1.6) in comparison to the two RMW experiments are shown in Table 3.10.

Table 3.10: The mean (± standard deviation) per-image-mean MRAE and AngE results of the baseline testing (single validation) and the two RMW experiments.

	Mean (± Std) Per-Image-Mean MRAE (%)		
	Baseline (SV)	**Original/RMW**	**RMW/RMW**
LR	6.39 (±2.90)	13.88 (±1.31)	9.16 (±0.92)
A+	3.81 (±1.99)	15.26 (±1.35)	9.41 (±0.90)
RBFN	2.09 (±1.11)	15.08 (±1.30)	9.52 (±1.21)
PR	1.98 (±1.11)	14.60 (±1.60)	8.55 (±0.88)
HSCNN-R	1.76 (±0.92)	12.96 (±1.50)	8.58 (±0.78)

	Mean (± Std) Per-Image-Mean AngE (deg)		
	Baseline (SV)	**Original/RMW**	**RMW/RMW**
LR	3.85 (±1.72)	7.83 (±0.77)	5.81 (±0.57)
A+	2.35 (±1.18)	8.30 (±0.79)	6.01 (±0.56)
RBFN	1.50 (±0.74)	8.21 (±0.64)	5.90 (±0.60)
PR	1.45 (±0.73)	9.01 (±0.93)	5.51 (±0.54)
HSCNN-R	1.36 (±0.64)	7.66 (±0.81)	5.76 (±0.53)

First, reviewing the baseline testing results, we see that algorithms with different model complexities also provide different level of performances. Indeed, in terms of MRAE, the DNN-based HSCNN-R method performs best overall, with an error of less than 1/3 the MRAE compared to linear regression (LR). Nevertheless, the performance gap compared to polynomial regression (PR) is much less. Similar trend is also observed in the AngE results, i.e., the DNN approach provides a modest uplift in performance compared to the best regression algorithm.

Let us now look at the **Original/RMW** results. Here we train on real images but then test on the unseen and unexpected RMW images. First, it is worth remarking that the recovery error for all algorithms is much worse. Also noteworthy is that all algorithms perform—more or less—equally well (actually, equally bad).

The results for the **RMW/RMW** experiment are when we retrain all the algorithms for our RMW images. Here, unsurprisingly, we get better results than the **Original/RMW** experiment, though the errors remain higher than the baseline. Curiously, however, the performance gap between the simplest LR and the HSCNN-R network still remains small. This means that even as we provide RMW scenes as training set for learning, the DNN still does not possess clear advantage against the simplest LR. And, the best algorithm overall, in terms of both MRAE and AngE, is now PR which is much less complicated than the DNN.

Evidently, we demonstrate that the model complexity does not help recovering the unseen and unexpected, randomly generated RMW images, both in terms of normal training and RMW training. In other words, under the worst-case imaging condition defined by RMW, we do not see the benefit of using the much more complex DNN method compared to the simple and primitive linear regression.

3.4 Summary

To summarise, in this chapter we examined the relative performance of SR models with different levels of complexities: regression, regression with clustering support, and Deep Neural Network (DNN).

First, we present a baseline experiment which benchmarks the methods using the same hyperspectral image database. It is shown that the DNN-based HSCNN-R method performs the best overall, but only by around 12% in terms of the MRAE metric compared to the best-performing regression: Polynomial Regression (PR).

We also show the results in terms of different spectral metrics. It is demonstrated that in most cases both settings do not change the methods' *rankings*, with one exception that is the mean per-image-99-percentile evaluation using the RMSE metric, where PR surpasses HSCNN-R. We hypothesised that this rank-performance flip is because the PR method is trained to minimise RMSE but the HSCNN-R minimises MRAE.

We also ran experiments to evaluate the relative performance of 10 cameras (with 10 different spectral sensitivities). The good news is that the rankings of the algorithms were shown not to depend on the camera used. However, care must be taken in SR experimentation. It is possible, for example, that a regression-based SR algorithm can deliver better spectral recovery than a DNN if the two methods use different cameras to capture their input RGBs.

Finally, we investigated the SR algorithms' performance under a worst-case real-world imaging condition we defined, called the Radiance Mondrian World (RMW) assumption. In RMW, the spatial patterns are random and unpredictable, and all natural radiances have the same chance to appear in each image, i.e., no one radiance is more likely to appear than another. With this assumed worst-case imaging condition, we show that all methods—regardless of their differences in complexity—degrade to broadly the same level of performance. We also found that even if we retrain all methods using an RMW training set, more complex algorithms still do not hold any advantage against even the simplest Linear Regression (LR) approach.

Chapter 4

On the Optimisation of Regression-Based Spectral Reconstruction

In Chapter 3, we saw that compared to the leading DNN method HSCNN-R, the polynomial-regression-based spectral reconstruction performs slightly worse in a regular benchmark test, and on par under the worst-case imaging condition.

In this chapter, we demonstrate that it is possible to alter—in the same systematic manner—all regression-based formulations of SR so that their recovery performance is improved.

4.1 Introduction

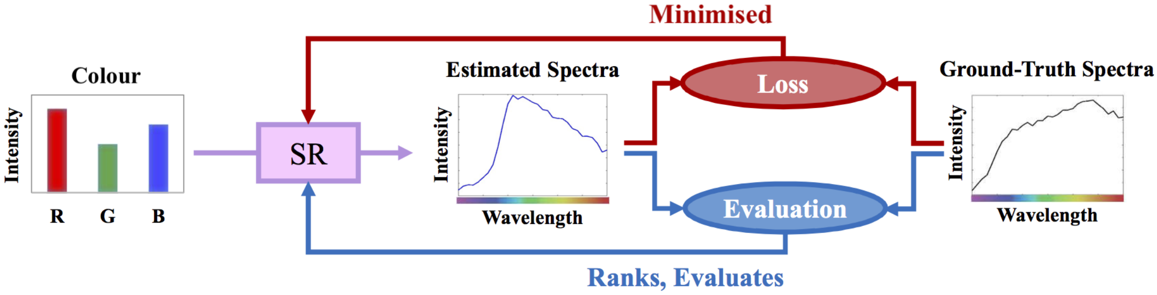


Figure 4.1: The standard spectral reconstruction training (**red arrows**) and evaluation scheme (**blue arrows**).

In Figure 4.1, 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 minimised. After the SR models are trained, we evaluate them based on a desired *evaluation* metric.

Ideally, the two metrics should *match* (i.e., the same or similar in nature). Indeed, a model that is optimised for one metric but evaluated by another will surely lead to sub-optimal results.

However, we noticed that in recent works [7; 8; 9], DNN-based models are most commonly evaluated and ranked by the Mean Relative Absolute Error (MRAE; Equation (3.1)). Most top DNN models are also designed to minimise MRAE directly [80; 53; 105; 7; 8]. But, all regressions used in SR are still optimised using the conventional least-squares minimisation, where the squared Root Mean Square Error (RMSE; Equation (3.2)) is the loss metric.

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

As a second contribution, we also propose a new way of regularising the regression-based spectral reconstruction. Most regressions are necessarily trained using a regularisation constraint [87], both to prevent overfitting [98] and to make the system equations more *stable* (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 regularised altogether—e.g., in [42; 1; 62; 25]. That is the regularisation constraint is effectively applied at the spectrum level. Yet, fundamentally, the MRAE metric measures the errors at individual spectral channels independently and then averages them to give the overall spectral error measure (see Equation (3.1)). We further show that in the conventional regression formulation the values at each spectral channel are also mapped separately from others. Similarly, we propose that the regularisation should also be carried out "per channel", i.e., to ensure optimised regularisation for each spectral channel independently. This new regularisation

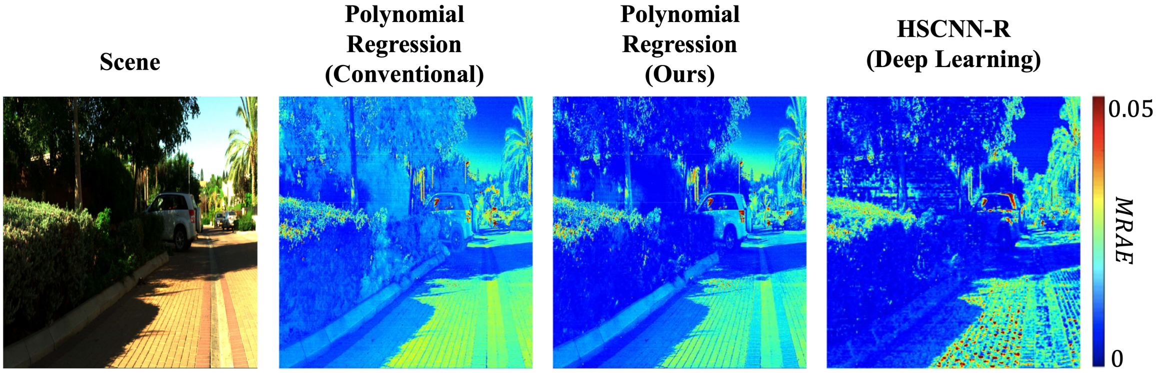


Figure 4.2: Example hyperspectral image reconstruction error heat maps (in MRAE) by the conventional Polynomial Regression, Polynomial Regression based on our new RELS method, and the DNN-based HSCNN-R.

strategy can be regarded as a standalone improvement for the conventional least-squares, but it is also adopted in both our RELS and RELAD formulations.

Combined, we find that training the simple regressions to minimise the same error as used in testing and adopting a per-channel regularisation approach lead to a significant uptick in performance. In particular, as shown in the example hyperspectral image reconstruction results in Figure 4.2, our RELS-based Polynonial Regression can now deliver more similar SR to the HSCNN-R approach.

4.2 Least-Squares Regression Revisited

As a recap to the regression methods (introduced in Section 2.3), generally, the goal of regression is to minimise the error of the following approximation:

$$\mathbf{MC} \approx \mathbf{H} \ , \tag{4.1}$$

where the columns of **H** are ground-truth radiance spectra, and the columns of **C** are simple features derived from the matching RGBs: in Linear Regression and A+, they are the RGBs themselves, whereas in Polynomial Regression and Radial-Basis-Function Network, their respective non-linear expansions are used on the RGBs to derive the features (see respectively Section 2.3.1 and 2.4.1 for more details). Matrix $\mathbf{M}$ is an $n \times p$ matrix with n being the spectral dimension and p the fixed length of the feature vectors, containing all learnable parameters of the regression mapping.

Conventionally, as introduced in Section 2.3.3, $\mathbf{M}$ is solved by least-squares minimisation with a ridge regularisation setting:

$$\min_{\mathbf{M}} \ ||\mathbf{MC} - \mathbf{H}||_2^2 + \gamma||\mathbf{M}||_2^2 \ . \tag{4.2}$$

Here, the first term in the loss function represents the least-squares minimisation, i.e., the minimisation of the sum of squared errors, and the second term bounds the ℓ_2 norm of the matrix $\mathbf{M}$. The competition in minimisation between the two terms is controlled by a single regularisation parameter γ.

4.3 Per-Channel Regularisation

The regularisation method used in the standard case (Equation (4.2)) takes place at the spectrum level with all spectral channels being regularised together. Here, we will argue—and develop the requisite mathematics—that the regularisation should be done per spectral channel.

To begin with, let us review again the goal of regression which is Equation (4.1). Here, without altering the goal, we can split the regression matrix $\mathbf{M}$ and the ground-truth spectral data $\mathbf{H}$ by rows:

$$\mathbf{MC} = \begin{bmatrix} \underline{\boldsymbol{m}}_1^\intercal \\ \underline{\boldsymbol{m}}_2^\intercal \\ \vdots \\ \underline{\boldsymbol{m}}_c^\intercal \\ \vdots \\ \underline{\boldsymbol{m}}_n^\intercal \end{bmatrix} \mathbf{C} \approx \begin{bmatrix} \underline{\boldsymbol{\eta}}_1^\intercal \\ \underline{\boldsymbol{\eta}}_2^\intercal \\ \vdots \\ \underline{\boldsymbol{\eta}}_c^\intercal \\ \vdots \\ \underline{\boldsymbol{\eta}}_n^\intercal \end{bmatrix} = \mathbf{H} \ . \tag{4.3}$$

Here, remember that the columns of $\mathbf{H}$ are individual radiance spectra, thus the values in, e.g., the cth row of $\mathbf{H}$, $\underline{\boldsymbol{\eta}}_c^{\mathsf{T}}$, are the cth-channel spectral intensities of all spectra in the database, and the length of $\underline{\boldsymbol{\eta}}_c^{\mathsf{T}}$ is N (i.e., the number of data points).

With this equivalent representation of the regression's mapping function, we see that regression-based SR is in actuality a collection of n independent *per-channel* regressions:

$$\underline{\boldsymbol{m}}_c^{\mathsf{T}}\mathbf{C} \approx \underline{\boldsymbol{\eta}}_c^{\mathsf{T}} \ ; \quad \text{for } c = 1, 2, \cdots, n \ . \tag{4.4}$$

Again, we emphasise that Equation (4.3) and (4.4) are both equivalent to the original formulation in Equation (4.1), only that they explicitly show that, by default, there is no "inter-channel" dependence exploited in regression. In other words, for all regression-based SR in the literature, it has been always the case that each row of $\mathbf{M}$ is only used for recovering the corresponding row of $\mathbf{H}$ while irrelevant to the recoveries for other spectral channels.

Curiously, as we solve for $\mathbf{M}$ using the standard minimisation in Equation (4.2), the strength of the penalty term, $\gamma||\mathbf{M}||_2^2$, is controlled by a single parameter γ. This means that all rows of $\mathbf{M}$—that is, the n separately-functioning $\underline{\boldsymbol{m}}_c^{\mathsf{T}}$—are regularised using the same γ parameter, despite the fact that each of them works independently of others. Essentially, by regularising $\mathbf{M}$ as a whole, we are "asserting" such an interdependence among channels.

From a mathematical viewpoint (regarding how the regression is formulated), we shall be able to select the best empirical γ parameter for each spectral channel separately. Following Equation (4.4), we reformulate the regularised least-squares minimisation (Equation (4.2)) in a per-channel fashion:

$$\min_{\underline{\boldsymbol{m}}_c^{\mathsf{T}}} \ ||\underline{\boldsymbol{m}}_c^{\mathsf{T}}\mathbf{C} - \underline{\boldsymbol{\eta}}_c^{\mathsf{T}}||_2^2 + \gamma_c||\underline{\boldsymbol{m}}_c^{\mathsf{T}}||_2^2 \ ; \quad \text{for } c = 1, 2, \cdots, n \ . \tag{4.5}$$

Here, the per-channel regularisation parameter γ_c is used specifically for regularising the regression of the cth spectral channel.

Similarly to the conventional least-squares whose closed-form solution exists (Equation (2.32)), Equation (4.5) can also be written in closed form [87]:

$$\underline{\boldsymbol{m}}_c^{\mathsf{T}} = \underline{\boldsymbol{\eta}}_c^{\mathsf{T}} \mathbf{C}^{\mathsf{T}} \left[\mathbf{C}\mathbf{C}^{\mathsf{T}} + \gamma_c \mathbf{I} \right]^{-1} . \tag{4.6}$$

4.3.1 Remark

To provide some intuition on why we think allowing the regularisation at different spectral channels to be different is important, we point out that it is *a priori* possible that the value distributions in different channels have different level of non-linear relationship with the input RGBs, and, e.g., if using a fixed high-order polynomial regression, the optimal amount of regularisation for fitting the different level of non-linearity in each channel will be different. Plus, the amount of noise in data can be wavelength dependent, which also calls for wavelength-dependent regularisation.

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 [91] on the recovered spectra, though we note that this assumption would be more important for the *reflectance recovery* since reflectances are usually intrinsically smooth, instead of the radiance spectra we are considering (real radiance spectra can be far from smooth because the illumination spectrum is part of the radiance, especially for indoor illuminations).

4.4 Minimising Relative Errors in Regression

So far, both the conventional and per-channel least-squares (LS) target to minimise a sum-of-squares loss in training. In this section we will argue that there is another type of error metric, namely the "relative errors", which should be better metrics to use for training and evaluating the SR algorithms.

4.4.1 RMSE versus MRAE Error Measures

Let us review the RMSE and MRAE metrics introduced in Section 3.1.4:

$$\text{RMSE}(\underline{\hat{\boldsymbol{h}}}, \underline{\boldsymbol{h}}) = \sqrt{\frac{1}{n}||\underline{\hat{\boldsymbol{h}}} - \underline{\boldsymbol{h}}||_2^2} \, , \tag{4.7}$$

$$\text{MRAE}(\underline{\hat{\boldsymbol{h}}}, \underline{\boldsymbol{h}}) = \frac{1}{n}\left|\left|\frac{\underline{\hat{\boldsymbol{h}}} - \underline{\boldsymbol{h}}}{\underline{\boldsymbol{h}}}\right|\right|_1 , \tag{4.8}$$

where $\underline{\hat{\boldsymbol{h}}} = \Psi(\underline{\boldsymbol{c}})$ is a simplified nomenclature for the SR estimation (Ψ denotes an SR algorithm), $\underline{\boldsymbol{h}}$ is the ground-truth spectrum, n is the number of spectral channels (i.e., the vector length of $\underline{\boldsymbol{h}}$ and $\underline{\hat{\boldsymbol{h}}}$), and the division in MRAE is component-wise. The division used in MRAE makes it a *relative* error, which refers to the type of errors measuring differences with respect to the ground-truth values.

Of course, the physical radiance spectrum (i.e., hyperspectral measurement) $\underline{\boldsymbol{h}}$ may have greater or lesser magnitudes depending on the imaging conditions. In reality, such exposure condition changes happen when the user changes the exposure time and/or aperture setting of the camera, the brightness of the prevailing illumination of the scene changes, and/or the same object is viewed in different parts of the image and subject to a changed level of highlight or shading. Under these circumstances, the measured spectrum is changed from $\underline{\boldsymbol{h}}$ to $k\underline{\boldsymbol{h}}$, where k is a constant scaling indicating the level of exposure change.

Correspondingly, an SR algorithm *might* suggest the same recovered spectrum scaled by the same constant, i.e., $k\underline{\hat{\boldsymbol{h}}}$. In this case, the algorithm predicts the same level of fitness of $\underline{\boldsymbol{h}}$ while **correctly** predicts the exposure scaling k. We note that, depending on the SR algorithms, this condition might not be met. However, this is definitely a preferred property of an SR algorithm, which will be discussed as a separate issue in Chapter 6.

Now, let us revert back to the RMSE and MRAE metrics. Given the same ground-truth and recovered spectrum before and after scaled by k, we get RMSE$(k\underline{\hat{\boldsymbol{h}}}, k\underline{\boldsymbol{h}}) = k$RMSE$(\underline{\hat{\boldsymbol{h}}}, \underline{\boldsymbol{h}})$. In the case of $k > 1$ (when both ground-truth and recovery get brighter), it seems the RMSE metric is "punishing" the algorithm for recovering the correct level of brightness. Arguably, the MRAE makes more sense as a performance metric, since for the same two cases it returns the same error: MRAE$(k\underline{\hat{\boldsymbol{h}}}, k\underline{\boldsymbol{h}}) =$ MRAE$(\underline{\hat{\boldsymbol{h}}}, \underline{\boldsymbol{h}})$.

This biased nature of RMSE evaluation can as well influence the training process of regressions. Indeed, we can expect that the standard least-squares minimisation can overestimate the sum-of-squares loss (i.e., squared RMSE) of the bright spectra and consequently place more importance on minimising their errors compared to the dim ones. Therefore, just as important as switching the use of RMSE to MRAE (or similar relative errors) in evaluation, it is of our interest to reformulate the regression to minimise a relative error loss function for unbiased SR training.

4.4.2 Relative Error Least Squares Regression

MRAE is an ℓ_1 error, whose minimisation can rarely be found in closed form. Therefore, our first attempt to relative-error-minimising regression is to make the regression minimise an ℓ_2 variant of relative error which leads to a closed-form minimisation.

We return back to the goal of regression (Equation (4.1)). Here, we can remodel the approximation—instead of least-squares—as the following minimisation:

$$\min_{\mathbf{M}} \left\| \frac{\mathbf{MC} - \mathbf{H}}{\mathbf{H}} \right\|_2^2 = \min_{\underline{\hat{\boldsymbol{h}}}_1, \underline{\hat{\boldsymbol{h}}}_2, \cdots, \underline{\hat{\boldsymbol{h}}}_N} \sum_{i=1}^{N} \left\| \frac{\underline{\hat{\boldsymbol{h}}}_i - \underline{\boldsymbol{h}}_i}{\underline{\boldsymbol{h}}_i} \right\|_2^2 \ , \ \underline{\hat{\boldsymbol{h}}}_i = \mathbf{M}\varphi(\underline{\boldsymbol{c}}_i) \ , \tag{4.9}$$

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 [5; 1]) is minimised. N is the number of spectra in the training set. $\varphi()$ is the feature expansion adopted by a particular regression. We call this new minimisation approach the Relative Error Least Squares (RELS).

Because in Equation (4.9) the divisions are component-wise, equivalently, we can rewrite the RELS minimisation in a per-channel fashion:

$$\min_{\underline{\hat{\boldsymbol{\eta}}}_1^\mathsf{T},\underline{\hat{\boldsymbol{\eta}}}_2^\mathsf{T},\cdots,\underline{\hat{\boldsymbol{\eta}}}_n^\mathsf{T}} \sum_{c=1}^{n} \left|\left|\frac{\underline{\hat{\boldsymbol{\eta}}}_c^\mathsf{T}-\underline{\boldsymbol{\eta}}_c^\mathsf{T}}{\underline{\boldsymbol{\eta}}_c^\mathsf{T}}\right|\right|_2^2 \ , \ \underline{\hat{\boldsymbol{\eta}}}_c^\mathsf{T} = \underline{\boldsymbol{m}}_c^\mathsf{T}\mathbf{C} \ . \tag{4.10}$$

Further, we can remove the summation symbol by regarding the minimisation in each channel separately (again, because of the lack of inter-channel independence), and get

$$\min_{\underline{\boldsymbol{m}}_c^\mathsf{T}} \left|\left|\frac{\underline{\boldsymbol{m}}_c^\mathsf{T}\mathbf{C}}{\underline{\boldsymbol{\eta}}_c^\mathsf{T}} - \underline{\mathbf{1}}^\mathsf{T}\right|\right|_2^2 \quad \text{for } c=1,2,\cdots,n \ , \tag{4.11}$$

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

To further simplify the nomenclature, let us define:

$$\mathbf{X}_c = \frac{\mathbf{C}}{\underline{\boldsymbol{\eta}}_c^\mathsf{T}} = \mathbf{C}\begin{bmatrix} 1/\eta_{c,1} & 0 & \cdots & 0 \\ 0 & 1/\eta_{c,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/\eta_{c,N} \end{bmatrix} \ , \tag{4.12}$$

where $\underline{\boldsymbol{\eta}}_c^\mathsf{T} = [\eta_{c,1},\eta_{c,2},\cdots,\eta_{c,N}]$. Using this nomenclature, Equation (4.11) is again rewritten into

$$\min_{\underline{\boldsymbol{m}}_c^\mathsf{T}} ||\underline{\boldsymbol{m}}_c^\mathsf{T}\mathbf{X}_c - \underline{\mathbf{1}}^\mathsf{T}||_2^2 \quad \text{for } c=1,2,\cdots,n \ . \tag{4.13}$$

Clearly, Equation (4.13) shows that RELS is in effect another per-channel least-squares problem (Equation (4.5)), but here we regress $\mathbf{X}_c$ to fit the row vector $\underline{\mathbf{1}}^\mathsf{T}$.

Of course, we need to regularise this minimisation by solving the following equation instead:

$$\min_{\underline{\boldsymbol{m}}_c^\mathsf{T}} ||\underline{\boldsymbol{m}}_c^\mathsf{T}\mathbf{X}_c - \underline{\mathbf{1}}^\mathsf{T}||_2^2 + \gamma_c||\underline{\boldsymbol{m}}_c^\mathsf{T}||_2^2 \quad \text{for } c=1,2,\cdots,n \ , \tag{4.14}$$

whose solution is written as [89; 87]

$$\underline{\boldsymbol{m}}_c^{\mathsf{T}} = \underline{\mathbf{1}}^{\mathsf{T}} \mathbf{X}_c^{\mathsf{T}} \left[\mathbf{X}_c \mathbf{X}_c^{\mathsf{T}} + \gamma_c \mathbf{I} \right]^{-1} \ . \tag{4.15}$$

This is the closed-form solution of RELS regression.

4.4.3 Relative Error Least Absolute Deviation Regression

Finally, let us consider to minimise MRAE directly. Analogous to Equation (4.9), we now wish to solve the ℓ_1 minimisation:

$$\min_{\mathbf{M}} \left|\left| \frac{\mathbf{MC} - \mathbf{H}}{\mathbf{H}} \right|\right|_1 = \min_{\hat{\underline{\boldsymbol{h}}}_1, \hat{\underline{\boldsymbol{h}}}_2, \cdots, \hat{\underline{\boldsymbol{h}}}_N} \sum_{i=1}^{N} \left|\left| \frac{\hat{\underline{\boldsymbol{h}}}_i - \underline{\boldsymbol{h}}_i}{\underline{\boldsymbol{h}}_i} \right|\right|_1 \ , \ \hat{\underline{\boldsymbol{h}}}_i = \mathbf{M} \varphi(\underline{\boldsymbol{c}}_i) \ . \tag{4.16}$$

Following the same derivation as Equation (4.9)–(4.14), we reach

$$\min_{\underline{\boldsymbol{m}}_c^{\mathsf{T}}} \ \left|\left| \underline{\boldsymbol{m}}_c^{\mathsf{T}} \mathbf{X}_c - \underline{\mathbf{1}}^{\mathsf{T}} \right|\right|_1 + \gamma_c \left|\left| \underline{\boldsymbol{m}}_c^{\mathsf{T}} \right|\right|_1 \quad \text{for } c = 1, 2, \cdots, n \ . \tag{4.17}$$

Notice that, here, not only do we minimise an ℓ_1 loss (MRAE), but also using an ℓ_1 regularisation penalty term. This refers to the LASSO regularisation [86] (see Section 2.3.3).

In the literature, regressions solved via an ℓ_1 minimisation is called the Least Absolute Deviation (LAD) [95; 19]. As here the MRAE we are minimising is a relative error, we call this new approach the Relative Error Least Absolute Deviation (RELAD).

Unlike RELS, RELAD does not have a closed-form solution. Linear programming [95; 92] is commonly used to find the globally optimal solution for small amount of data. However, its requirement for computational resources can drastically increase for large amount of data (e.g., in our application). Alternatively, the Iterative Reweighted Least Squares (IRLS) [19] algorithm is more appropriate and is thus used here. The IRLS process approaches RELAD minimisation by repeatedly solving Weighted Least Squares (WLS) [18] while updating the

Algorithm 1 Solving RELAD regression (Equation (4.17)) 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: $\underline{\boldsymbol{\delta}}^{(0)} = \inf$    ▷ $N$-vector; Placeholder to record the per-sample absolute losses
3: $t = 0$
4: repeat
5:     $t = t + 1$
6:     $\underline{\boldsymbol{m}}_c^{\mathsf{T}} = \underline{\mathbf{1}}^{\mathsf{T}}\mathbf{W}^{(t-1)}\mathbf{X}_c^{\mathsf{T}}\left[\mathbf{X}_c\mathbf{W}^{(t-1)}\mathbf{X}_c^{\mathsf{T}} + \gamma_c\widetilde{\mathbf{W}}^{(t-1)}\right]^{-1}$    ▷ Closed-form WLS solution
7:     $\underline{\boldsymbol{\delta}}^{(t)} = \left[\mathrm{abs}(\underline{\boldsymbol{m}}_c^{\mathsf{T}}\mathbf{X}_c - \underline{\mathbf{1}}^{\mathsf{T}})\right]^{\mathsf{T}}$    ▷ Updated absolute losses
8:     $\hat{\sigma} = \dfrac{\mathrm{median}\big(\underline{\boldsymbol{\delta}}^{(t)}\big)}{0.6745}$    ▷ An estimated standard deviation of the losses
9:     $\mathbf{W}^{(t)} = diag\Big(\dfrac{\hat{\sigma}}{\max(\underline{\boldsymbol{\delta}}^{(t)}, \epsilon_1)}\Big)$
10:    $\widetilde{\mathbf{W}}^{(t)} = diag\Big(\dfrac{1}{\max(\mathrm{abs}(\underline{\boldsymbol{m}}_c), \epsilon_1)}\Big)$
11: until $\left|\mathrm{mean}\big(\underline{\boldsymbol{\delta}}^{(t)}\big) - \mathrm{mean}\big(\underline{\boldsymbol{\delta}}^{(t-1)}\big)\right| < \epsilon_2$ or $t \geq T$
12: return $\underline{\boldsymbol{m}}_c^{\mathsf{T}}$
```

weights on every iteration depending on the losses and mapping functions obtained in the previous iteration, until the solution converges.

The detailed algorithm we used is given in Algorithm 1. The abs() (taking absolute values) and divisions are component-wise to the vectors. $\max(\underline{\boldsymbol{v}}, \epsilon)$ clips the lowest value in $\underline{\boldsymbol{v}}$ at ϵ. The median() and mean() functions respectively calculate the median and mean of the vector components.

In Step 1 to 3, we initialise the parameters we are to update in every iteration, including the weights used in WLS, the per-sample absolute losses, and the iteration number. We put a superscript $^{(t)}$ to indicate the iteration in which the parameters are derived. Then, we iteratively solve $\underline{\boldsymbol{m}}_c^{\mathsf{T}}$ using closed-form WLS (Step 6), with the weights updated according to Step 9 and 10. Here, $\hat{\sigma}$ is the standard deviation of the losses estimated from their Median Absolute Deviation (MAD), which is an estimation of scale commonly used in robust statistics [73; 19]. ϵ_1 is a small number (we set $\epsilon_1 = 10^{-6}$) which stabilises the algorithm by preventing divisions of too small numbers. Finally, the stopping criteria are defined in Step 11: when the change of *mean losses* (here, i.e., the channel-wise relative error in the training set) in two consecutive iterations is less than $\epsilon_2 = 0.00005$ or when the iteration t reaches $T = 20$. We set this particular ϵ_2 tolerance because the MRAE results presented in this thesis and in recent literature are rounded to 4 decimal places, whereas the setting of T is where we observed convergence for most of our experiments.

4.5 Experiments and Results

Table 4.1: List of minimisation approaches.

Approach	**Per-Channel Regularisation**	**Loss Metric**
LS	✗	squared RMSE
Per-Channel LS (LS$^{\text{pc}}$)	✓	squared RMSE
RELS	✓	squared relative-RMSE
RELAD	✓	MRAE

In this chapter we proposed 3 new optimisation approaches to regression-based SR, including the conventional least squares with per-channel regularisation (LS$^{\text{pc}}$), RELS and RELAD. Table 4.1 shows the comparison of the conventional least squares (LS) approach and our new methods. Here, we present results for the regression methods—Linear Regression (LR), Polynomial Regression (PR), Radial-Basis-Function Network (RBFN) and A+ Sparse Coding (A+)—optimised via the 4 minimisation approaches in Table 4.1.

The experiments are conducted using the cross validation (CV) methodology (Section 3.1.3). The camera spectral sensitivities used to simulate the ground-truth RGB images are the CIE 1964 colour matching functions (CMF; Section 3.2) [24]. The baseline HSCNN-R performance will also be provided here to ease comparison.

Table 4.2: The cross-validated mean per-image-mean and per-image-99-percentile (Pt.) MRAE for regressions trained via LS, LS$^{\text{pc}}$, RELS and RELAD minimisation approaches. The best approach for each regression method is shown in bold and underlined. The baseline result of HSCNN-R (the reference DNN model) is given in the last row.

	Mean Per-Image-Mean MRAE (%)				**Mean Per-Image-99-Pt. MRAE (%)**			
	LS	**LS$^{\text{pc}}$**	**RELS**	**RELAD**	**LS**	**LS$^{\text{pc}}$**	**RELS**	**RELAD**
LR	6.24	6.05	5.63	**5.36**	16.95	17.79	**14.09**	16.56
A+	3.87	3.75	3.60	**3.49**	15.26	14.28	**13.50**	14.02
RBFN	2.06	2.03	**1.98**	2.03	7.89	7.93	**7.40**	7.99
PR	1.95	1.93	**1.88**	1.92	7.10	7.09	**7.04**	7.34
HSCNN-R	1.73				6.53			

Table 4.3: The paired two-sample Student's t-test scores of the mean per-image-mean MRAE results. For each regression model, the best minimisation approach (the one has the lowest number in Table 4.2) is tested against each of the other approaches.

		Student's t-Test Score of Mean Per-Image-Mean MRAE			
	Best Approach	**Best vs. LS**	**Best vs. LS$^{\text{pc}}$**	**Best vs. RELS**	**Best vs. RELAD**
LR	RELAD	9.46	9.54	4.18	N/A
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

4.5.1 Mean and Worst-Case Performance

In Table 4.2, we present the mean (mean per-image-mean) and worst-case (mean per-image-99-percentile) MRAE statistics.

Let us first look at the numbers in the first row, which are results of variations of LR. The mean results (left table) show that LR trained using all of our three new approaches outperform the conventional LS, among which the RELAD method performs the best—returning 14% lower MRAE compared to LS. On the right side of Table 4.2 (headlined "Mean Per-Image-99-Pt. MRAE"), we see that RELS-based LR provides 17% lower worst-case MRAE compared to using the standard LS.

Similarly to our analysis for LR, observing all other regression models, we found that the best minimisation criterion in terms of mean MRAE is either RELS or RELAD depending on the model, while for all regressions RELS delivers the best worst-case performance.

To further examine the robustness of our best minimisation approach (RELS or RELAD), we conduct the paired Student's t-test [81] (used in Section 3.2) between the best approach and others. The t-scores are presented in Table 4.3. Here, unlike in Section 3.2, we adopt a cross-validation (CV) setup where all 200 images are used as testing images, which means the t-test here are calculated based on the 200 observed results (as opposed to 50 for the single-validation setup used in Section 3.2).

For example, let us look at the top-left number of Table 4.3, 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 minimisation. In the same row, we also see the t-test scores comparing the best approach with LS^{pc} and RELS, respectively, but not RELAD (because RELAD is the best approach itself).

Considering we have 200 observed values ($200 - 1 = 199$ degrees of freedom) and the one-sided hypothesis, a 5% level of significance corresponds to a t-score of 1.65 [49]. 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 minimisation criterion can consistently deliver the best per-image-mean MRAEs of all approaches.

Still, it is somewhat counterintuitive that RELAD performs worse than RELS in some cases. Indeed, as RELAD directly minimises 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 the ℓ_1 Least Absolute Deviation problems [75; 35; 19]. Another possible cause which might result in suboptimal optimisation 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. Indeed, it is well known that ℓ_1 minimisations tend to be less sensitive to outliers compared to their ℓ_2 counterparts [95; 19]. In other words, in RELAD, these 99-percentile pixels might be treated as outliers during training—i.e., minimisation 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 [7] 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 optimised 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 optimise these parameters (the ICVL database [5] used in our study and in NTIRE 2018 challenge [7] is already one of the largest available hyperspectral image databases so far).

4.5.2 Computational Time

Again, let us examine the computational time of our new approaches (Table 4.4). We calculate the training time as the average time used for each of the 4 experiments in our cross validation, and the reconstruction time as the average time used to reconstruct each tested image.

Table 4.4: The training and reconstruction time measurements for the LS, LS$^{\text{pc}}$, RELS and RELAD variants of the regression-based SR algorithms. The HSCNN-R time consumptions are also supplied in the bottom row as a reference. The same hardware specification is used as in Section 3.1.6.

	Training Time				**Reconstruction Time**			
	(Per Cross-Validation Trial)				**(Per Image)**			
	LS	**LS$^{\text{pc}}$**	**RELS**	**RELAD**	**LS**	**LS$^{\text{pc}}$**	**RELS**	**RELAD**
LR	6.7 min	6.1 min	6.2 min	36.0 h	0.031 s	0.032 s	0.032 s	0.033 s
A+	26.9 min	52.8 min	54.6 min	46.2 h	13.7 s	13.7 s	13.7 s	13.7 s
RBFN	1.0 h	1.0 h	1.2 h	36.8 h	3.1 s	3.1 s	3.1 s	3.1 s
PR	15.1 min	15.4 min	42.2 min	44.6 h	6.0 s	5.9 s	5.9 s	6.0 s
HSCNN-R	35.5 h (with GPU)				13.0 s (with GPU)			

First, we look at the training time (the left side of Table 4.4). We see that regardless of the regression model, the closed-form LS, LS$^{\text{pc}}$ 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 minimisation 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 4.4), 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 minimisation approach. This result makes sense because, after all, no matter how we optimised the regression matrix, the reconstruction procedure is the same. Consequently, in the case that longer "offline" training time is permitted, the RELAD-optimised regression can still support fast reconstruction just like LS, LS$^{\text{pc}}$, and RELS.

4.5.3 Brightness Dependence

Let us further investigate the performance discrepancy within an image. In Figure 4.3, we show for each regression approach the MRAE recovery error map of a given example scene.

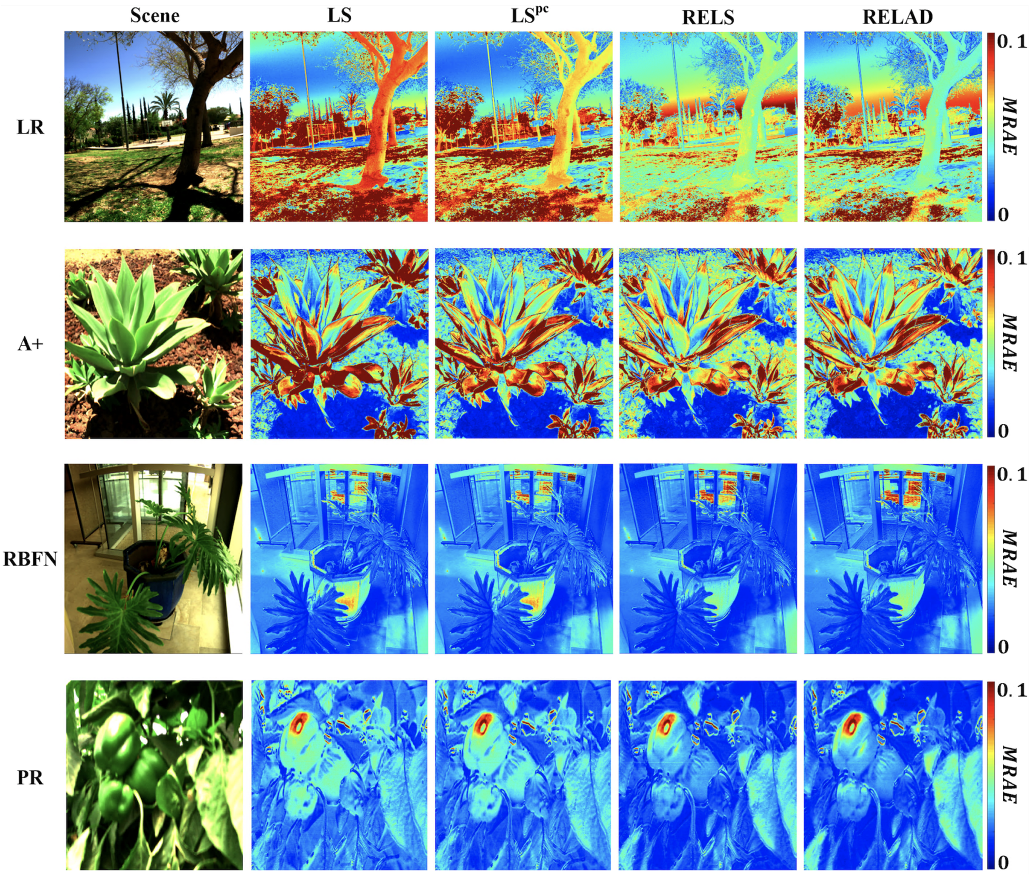
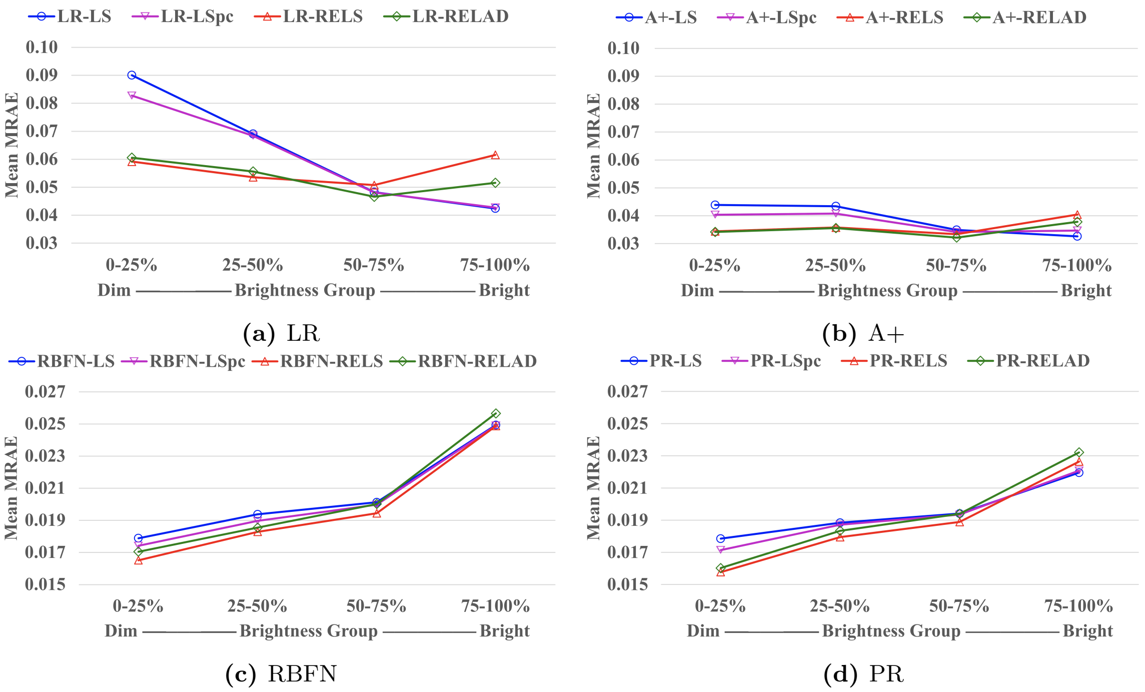


Figure 4.3: The MRAE error heat maps for the considered regression models optimised for LS, LS^{pc}, RELS, and RELAD criteria.

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, 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. 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 minimisation (see Section 4.4.1).



(a) LR **(b)** A+

(c) RBFN **(d)** PR

Figure 4.4: 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**,**b**) the Mean MRAE (the vertical axes) is shown between the interval of $[0.03, 0.10]$, while in plot (**c**,**d**) it is shown between $[0.015, 0.027]$.

In Figure 4.4, we illustrate how the mean MRAE performance of each regression approach is related to the pixels' *brightness*—i.e., 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.

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.

4.6 Summary

Regression-based spectral reconstruction has 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. Most recent benchmarks adopt the Mean Relative Absolute Error (MRAE) as the standard metric for evaluation and ranking. Following this trend, recent DNN models are also designed to directly minimise this metric. Comparatively, all regressions are still trained based on the least-squares minimisation which does not suggest a minimised MRAE result. The problem is further compounded by the sub-optimal regularisation setting where all independently-estimated spectral channels are regularised by the same single penalty parameter.

In this chapter, we developed new regression approaches that minimise relative errors and are regularised per spectral channel, including the closed-form Relative-Error Least Squares (RELS) and the Relative-Error Least Absolute Deviation (RELAD) approach (which directly minimises MRAE and was solved by an iterative method). Our results showed that the new minimisation 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 8.7% under the MRAE evaluation.

Chapter 5

Advancing Sparse-Coding-Based Spectral Reconstruction

In Chapter 4, we developed the RELS-based Polynomial Regression (PR), and henceforth we will abbreviate this method to “PR-RELS”. There we show that PR-RELS effectively narrows the gap between PR and the leading DNN, HSCNN-R, from the original 12% to 8.7%.

In this chapter, we are going to explore the clustering-based SR approach, especially evolving from the A+ sparse coding method [1]. In A+, the RGB space is divided into neighbourhoods, and the RGBs in different neighbourhoods are regressed separately. Here, we are to propose modifications on how the local neighbourhoods are defined, which will lead to significant improvements on its upper-bound performance. Then, we propose an approximated model, named “A++”, which effectively *approaches* the upper bound, delivering the state-of-the-art performance. As A++ is a pixel-based algorithm, and as its performance is in the same ballpark as a leading DNN approach, our result, effectively, calls into question the widely held belief that patch-based algorithms outperform their pixel-based counterparts.

5.1 A+ Sparse Coding Revisited

In A+ [1], the K-SVD algorithm [3] is used to find a set of K representative spectra out of N training spectra:

$$\mathbf{D}^h = K\text{-SVD}(\mathbf{H}) = [\underline{\boldsymbol{h}}^1, \underline{\boldsymbol{h}}^2, \cdots, \underline{\boldsymbol{h}}^j, \cdots, \underline{\boldsymbol{h}}^K] \ , \tag{5.1}$$

where the columns of $\mathbf{H}$ are individual training spectra. This dictionary is optimised (by K-SVD) such that all training spectra in $\mathbf{H}$ can be represented as linear combinations of the representatives in $\mathbf{D}^h$ with minimal errors.

An important assumption called "neighbour embedding" is adopted which entails that the neighbours in the spectral space are also neighbours in the RGB space (and vice versa) [1; 88]. Based on this assumption, A+ projects the dictionary of representative spectra, $\mathbf{D}^h$, to the RGB space following the colour image formulation formula (Equation (2.37)), deriving a K-member dictionary of RGBs, $\mathbf{D}^c$. Then, fixed-sized clusters are defined in the RGB space using $\mathbf{D}^c$ instead of $\mathbf{D}^h$, by finding the M closest RGBs among the training data points around each member of $\mathbf{D}^c$. Let us use the jth member of $\mathbf{D}^c$, $\underline{\boldsymbol{c}}^j$, as an example. A bespoke linear regression SR mapping is trained using the M training RGBs closest to $\underline{\boldsymbol{c}}^j$ and their corresponding ground-truth spectra. This mapping is used, in inference, only by query RGBs that are also nearby $\underline{\boldsymbol{c}}^j$ (i.e., whose closest $\mathbf{D}^c$ member is $\underline{\boldsymbol{c}}^j$).

5.2 The Oracle A+

Clearly, in A+, the neighbourhoods (clusters) are defined in the RGB space, even though the cluster *centres*—the representatives—are optimised in the spectral space and later projected to RGB. With the neighbour embedding assumption, clustering in the RGB space is equivalent to clustering in the spectral space. Indeed, if the RGB neighbours were also neighbours in the spectral space, an RGB cluster could also define a neighbourhood in the spectral space. However, this assumption is not always true. Based on the metamerism effect, two very different spectra can appear the same in colour [29; 99], which means they are RGB neighbours but may not be spectral neighbours.

Now, let us actually consider clustering data in the spectral space (without assuming clustering in RGB space equals clustering in spectral space). In an extreme case where each spectral data point forms its own cluster, if we could always select the correct clusters, we would always reach the correct answers (we get **zero** errors). Next, we reduce the number of clusters through the K-SVD representation training (Equation (5.1)) and then model the RGB-to-spectrum mapping at each spectral neighbourhood as a linear regression SR (i.e., the same A+ processes but operated in the spectral space instead of the RGB space). Of course, as long as we do not reduce the number of clusters by too much, it is still safe to say the predictions by local linear regressions should not be too far off the correct ones (because locally all spectra are already very similar).

The main problem of this setup is that we do not have means to always select the correct clusters for all query data, because the ground-truth spectra are unknown in practice. But, we can still study the *upper-bound* performance of this setup by assuming this knowledge to be known. We call this the “oracle solution” of A+ (or Oracle A+ in short). The point of formulating such an oracle solution is to firstly study the full potential of the A+ configuration, as we could later seek to approach this upper-bound performance.

A comparison of the training and reconstruction schemes of the original and the oracle A+ is given in Figure 5.1. As we can see, in the oracle solution (right of Figure 5.1), in each local linear regression training, the spectra are actual neighbours (as opposed to the original A+ where the RGBs are—but spectra might not be—neighbours). Correspondingly, in reconstruction, we find the closest dictionary member to the ground-truth spectrum which determines which local linear regression to use for a given input RGB. And, again, in Oracle A+ we assume the knowledge of the spectral output, which is not available in practice.

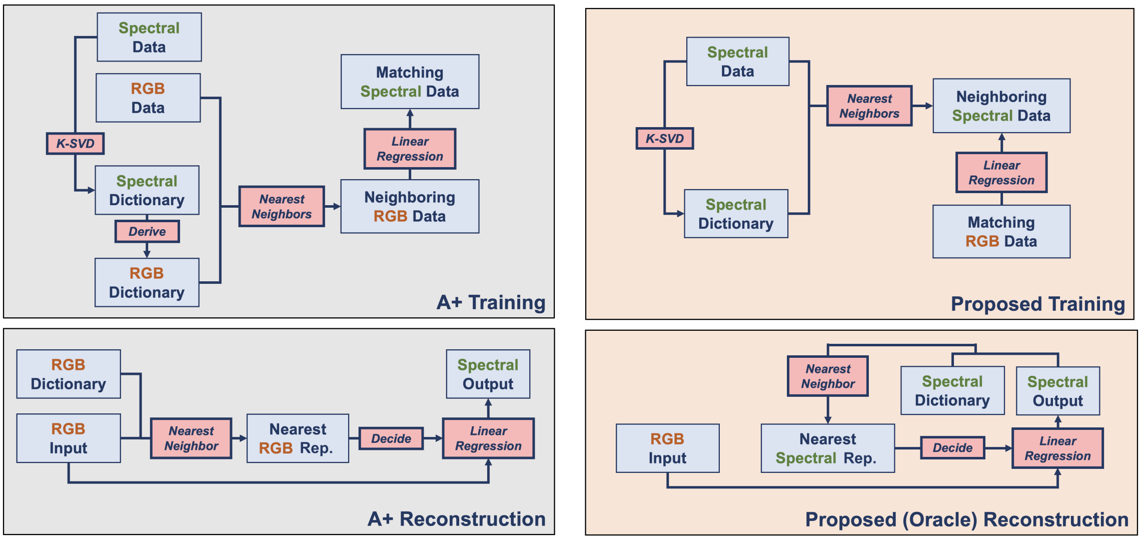


Figure 5.1: The training and reconstruction schemes of the original A+ (**left**) and our proposed oracle setup for A+ (**right**). "Rep." is short for the "representatives" included in the dictionary.

5.2.1 Oracle A+ versus DNN

Following the 4-fold cross validation setting introduced in Section 3.1.3, we compare the performance of the original A+, Oracle A+ and the DNN-based HSCNN-R. Note that we consider the conventional least-squares method (not the RELS or RELAD methods we developed in Chapter 4) for the local linear regression training in A+ and Oracle A+. We also keep the same hyperparameters of A+ used in Oracle A+, including the number of clusters $K = 1024$ and the number of nearest neighbours in each fixed-sized cluster $M = 8192$.

In Table 5.1, we show the mean per-image-mean and per-image-99-percentile MRAE results. Visualised error maps for one example testing image are also shown in Figure 5.2.

Evidently, we show that the upper-bound performance of A+ (i.e., the performance of Oracle A+) is much better than HSCNN-R in terms of the mean performance. Oracle A+ also have much better per-image 99 percentile results compared to the original A+, though slightly falls short compared to HSCNN-R.

Table 5.1: Comparing the mean hyperspectral image reconstruction accuracy delivered by A+, Oracle A+, and HSCNN-R, in terms of per-image-mean and per-image-99-percentile (Pt.) MRAE. Best results are shown in bold and underlined.

Method	**Mean Per-Image-Mean MRAE (%)**	**Mean Per-Image-99-Pt. MRAE (%)**
A+	3.81	15.52
HSCNN-R	1.73	<u>**6.53**</u>
Oracle A+	<u>**1.49**</u>	7.54

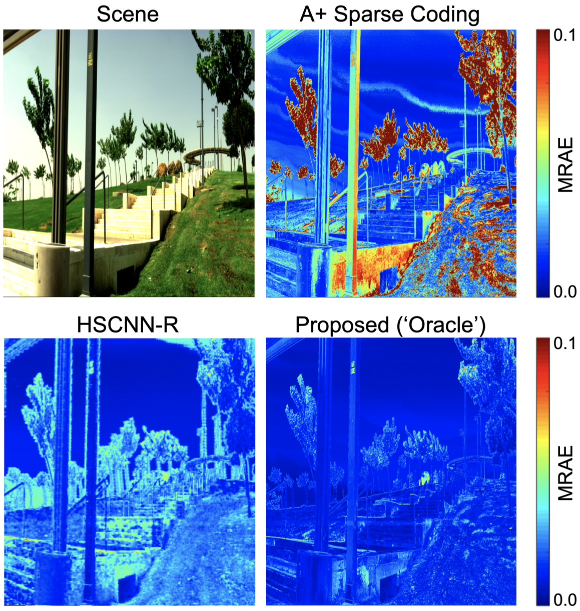


Figure 5.2: Reconstruction error maps of an example scene in MRAE.

We note that the Oracle A+ here represents the upper-bound of A+ when the specific hyperparameters, i.e., K and M, are adopted. It is possible that the oracle performance can be further advanced if these two parameters are tuned.

5.3 A++ Spectral Reconstruction

To make the Oracle A+ idea feasible in practice, we are to estimate the spectral neighbourhood labels from the RGBs. We can, of course, use another SR method that provides primary estimates of the spectra from the RGBs and use this estimated information to determine their spectral neighbourhoods. Then, we apply the local linear regression SR maps attached to these spectral neighbourhoods to the RGBs to deliver the final spectral recovery.

This procedure might seems convoluted. Indeed, as we use the primary SR method we would have reconstructed the spectra already. Yet, the proposed sparse coding procedure is still worth doing because, as presented in Section 5.2.1, the Oracle A+ defines an upper-bound performance much better than the leading DNN method and certainly even better than all the regression-based methods. This means whichever primary SR algorithm we use, we can potentially improve its original performance using the additional sparse coding steps (by “approaching” the upper bound defined by the Oracle A+).

To further refine the overall performance of the framework, we also run the K-SVD algorithm on the primary spectral estimates instead of the ground-truth spectra. That is we define spectral neighbourhoods among the primary estimates while the corresponding ground-truths might not be actual neighbours (but, of course, the local linear regressions are still aiming at predicting the ground-truth spectra, not the primary estimates which are only used to estimate the neighbourhood labels). This allows the algorithm to, potentially, correct the estimate when the primary SR algorithm may occasionally give wrong predictions of the actual, ground-truths’, spectral neighbourhoods.

Finally, our method—called the “A++”—is summarised in Table 5.2. We will dedicate the rest of this section to providing the specifics of our A++ implementation.

5.3.1 Primary SR Algorithm

The choice for our primary SR algorithm is not *a priori* fixed. For example, we may simply use the state-of-the-art DNN as the primary SR. Nevertheless, regarding that the Oracle A+ itself does not involve any implementation of the complex DNN architecture, we wish to take this chance to see if we could formulate an SR method that is completely without DNN implementation yet performs better than the DNNs. Hence, here, we choose the PR-RELS method (i.e., Polynomial Regression with RELS minimisation; Section 4.4.2) to

Table 5.2: A summary of the training and testing (reconstruction) process of A++.

Training steps	**Testing (reconstruction) steps**
1. Obtain primary SR estimates of all training RGBs	1. Obtain the primary SR estimate of each testing RGB
2. Run K-SVD on the primary estimates, returning K representative primary estimates	2. Find the closest representative (out of the K representatives) of each primary estimate
3. Around each representative, find M RGBs in the training set whose primary estimates are the closest	3. Get the trained local linear SR map associated with this closest representative
4. Train a linear SR map associated with this representative using the found M RGBs and their ground-truth spectra	4. Apply this map to the testing RGB to reconstruct its spectrum

be the primary SR algorithm. We note that PR-RELS is currently the best regression-based method considering the methods in the literature and in the previous chapter of this thesis.

As a recap, in PR-RELS, we find a global linear transformation matrix, $\mathbf{M}$, which maps the polynomial-expanded RGBs to spectral estimates that approximate the ground-truth spectra (Section 2.3.1):

$$\mathbf{M}\mathbf{C}^{\varphi} = \widehat{\mathbf{H}} \approx \mathbf{H} \ , \tag{5.2}$$

where the matching columns of $\mathbf{C}^{\varphi}$, $\widehat{\mathbf{H}}$ and $\mathbf{H}$ are respectively the polynomial expansions of the training RGBs, the PR-RELS primary estimates, and the ground-truth training spectra. Considering the 6th-order polynomial expansion (as we have always considered for polynomial regressions), $\mathbf{M}$ is an $n \times 83$ matrix, where n is the dimension of the spectral vector, and $n = 31$ for the ICVL hyperspectral database [5] we use in this thesis.

The RELS minimisation solves $\mathbf{M}$ by minimising:

$$\mathbf{M} = \min_{\mathbf{M}} \left|\left| \frac{\mathbf{M}\mathbf{C}^{\varphi} - \mathbf{H}}{\mathbf{H}} \right|\right|_2^2 , \tag{5.3}$$

where the division is component-wise to the matrices. For the closed-form solution of Equation (5.3) and its regularisation setting, readers are referred to Section 4.4.2. Here we assume PR-RELS has been pre-trained with the same set of training data prior to our sparse coding process.

5.3.2 Clustering the Primary Estimates

In our new clustering setup, we find an alternative dictionary of K representatives among $\widehat{\mathbf{H}}$, i.e., the PR-RELS primary estimates from the training RGBs:

$$\mathbf{D}^{\hat{h}} = K\text{-SVD}(\widehat{\mathbf{H}}) = [\widehat{\underline{\boldsymbol{h}}}^1,\ \widehat{\underline{\boldsymbol{h}}}^2,\ \cdots\ ,\ \widehat{\underline{\boldsymbol{h}}}^j,\ \cdots\ ,\ \widehat{\underline{\boldsymbol{h}}}^K]\ . \tag{5.4}$$

Here, the superscript ^{j} indexes the representatives.

Like in the original A+ where fixed-sized local clusters are defined as the M training RGBs closest to each RGB representative (Section 2.4.2; Equation (2.40)), here, we find the M RGBs whose PR-RELS primary estimates are closest to each primary estimate representative in our new $\mathbf{D}^{\hat{h}}$ dictionary. We write:

$$\widehat{\mathbf{H}}^j = \text{Prox}^M(\widehat{\mathbf{H}}, \widehat{\underline{\boldsymbol{h}}}^j) = [\widehat{\underline{\boldsymbol{h}}}_1^j,\ \widehat{\underline{\boldsymbol{h}}}_2^j,\ \cdots\ ,\ \widehat{\underline{\boldsymbol{h}}}_i^j,\ \cdots\ ,\ \widehat{\underline{\boldsymbol{h}}}_M^j]\ , \tag{5.5}$$

where the Prox^M function selects M columns in $\widehat{\mathbf{H}}$ that are closest to the jth representative $\widehat{\underline{\boldsymbol{h}}}^j$ in terms of Euclidean distances of *normalised* vectors (i.e., both $\widehat{\underline{\boldsymbol{h}}}^j$ and columns of $\widehat{\mathbf{H}}$ are normalised when calculating distances), and the columns of $\widehat{\mathbf{H}}^j$, i.e., $\widehat{\underline{\boldsymbol{h}}}_i^j$ for $i = 1, 2, 3, \cdots, M$, are the found M neighbours of $\widehat{\underline{\boldsymbol{h}}}^j$.

5.3.3 Local Linear Regressions

With respect to $\widehat{\mathbf{H}}^j$, column by column, we can trace back to the RGBs these primary spectral estimates are recovered from. Then, we can also find the corresponding ground-truth spectra of these RGBs. Respectively, we will use $\mathbf{C}^{j'}$ and $\mathbf{H}^{j'}$ to denote the sets of RGBs and ground-truth spectra in the jth primary spectral estimate neighbourhood. The columns of $\mathbf{C}^{j\prime}$, $\mathbf{H}^{j\prime}$ and $\widehat{\mathbf{H}}^{j}$ are matching RGBs, ground-truth spectra, and PR-RELS primary spectral estimates of the training data. Clearly, here, the locality of data is defined by $\widehat{\mathbf{H}}^{j}$, and $\mathbf{C}^{j\prime}$ and $\mathbf{H}^{j\prime}$ are merely the column-by-column correspondences of $\widehat{\mathbf{H}}^{j}$.

We use $'$ in $\mathbf{C}^{j\prime}$ and $\mathbf{H}^{j\prime}$ to distinguish them from the $\mathbf{C}^{j}$ and $\mathbf{H}^{j}$ used in the original A+ formulation in Section 2.4.2; Equation (2.40). In the original A+, $\mathbf{C}^{j}$ contains actual RGB neighbours, whereas in A++, the RGBs in the columns of $\mathbf{C}^{j\prime}$ are not necessarily neighbours. In other words, different sets of data are included in $\mathbf{C}^{j\prime}$ and $\mathbf{H}^{j\prime}$ compared to $\mathbf{C}^{j}$ and $\mathbf{H}^{j}$.

Then, the local linear regressions operate identically to the original A+ (see Equation (2.44)):

$$\mathbf{M}^{j\prime}\underline{\boldsymbol{c}}_i = \mathbf{H}^{j\prime}\mathbf{C}^{j\prime\mathsf{T}}[\mathbf{C}^{j\prime}\mathbf{C}^{j\prime\mathsf{T}} + \gamma\mathbf{I}]^{-1}\underline{\boldsymbol{c}}_i \approx \underline{\boldsymbol{h}}_i \ , \tag{5.6}$$

where $\underline{\boldsymbol{c}}_i$ is a query RGB in the testing step whose primary estimate's closest representative in $\mathbf{D}^{\widehat{h}}$ is $\widehat{\underline{\boldsymbol{h}}}^{j}$, $\mathbf{M}^{j\prime}$ is the local linear regression matrix attached to $\widehat{\underline{\boldsymbol{h}}}^{j}$ that can be determined in closed form with $\mathbf{H}^{j\prime}$ and $\mathbf{C}^{j\prime}$ in the training stage, and $\underline{\boldsymbol{h}}_i$ is the ground-truth testing spectrum we are to estimate from $\underline{\boldsymbol{c}}_i$. The $\gamma\mathbf{I}$ term ($\mathbf{I}$ is the 3×3 identity matrix) originates from the ridge regularisation process, which is detailed in Section 2.3.3.

5.4 Experiments

In this section, we will compare our new A++ method with HSCNN-R, and also PR-RELS and A+, the two methods A++ is built upon.

Like the studies in previous chapters, we use the ICVL hyperspectral image dataset [5] as ground-truth spectral database, and the CIE 1964 colour matching functions ([24]; Section 3.2) are used as the camera spectral sensitivity functions for ground-truth RGB image formation.

Table 5.3: The single-validated mean per-image-mean-MRAE performance in relation to the number of clusters (K) and the size of each cluster (M) used in our A++ method. The best result for each factor (while fixing the other factor) is shown in bold font and underlined.

K	1024	2048	4096	**<u>8192</u>**	10240
M (fixed)			8192		
MRAE (%)	1.88	1.82	1.78	**<u>1.76</u>**	1.78

K (fixed)			8192		
M	512	**<u>1024</u>**	2048	4096	8192
MRAE (%)	1.70	**<u>1.69</u>**	1.70	1.72	1.76

5.4.1 Tuning the A++ Architecture

According to Section 5.3.2, there are 2 major hyperparameters in A++ that dictate the clustering outcome, which are K, the number of clusters, and M, the number of training data included in each cluster. In the original A+ model [1], and also our test on Oracle A+ (Section 5.2.1), $(K, M) = (1024, 8192)$, and yet this might not be the best setting for A++. So, we are to re-determine both factors.

We started with fixing $M = 8192$ and search for the best K setting (top table of Table 5.3). We used the single validation evaluation setup (Section 3.1.3) and calculated the mean per-image-mean MRAE results, finding out that $K = 8192$ returns the lowest testing error. Then, we fixed $K = 8192$ and searched for the best setting for M. As shown in the lower table of Table 5.3, we see $M = 1024$ delivers the best performance.

Hence, we will use $(K, M) = (8192, 1024)$ for our A++ implementation. All other hyperparameters are kept the same as the original A+ detailed in Section 3.1.5.

5.4.2 Results

Like in previous chapters, we present the mean per-image-mean and per-image-99-percentile MRAE performance of all considered models in Table 5.4. This evaluation follows the 4-fold

Table 5.4: The cross-validated mean per-image-mean and per-image-99-percentile hyperspectral image reconstruction accuracy in terms of MRAE, comparing the pixel-based A+, PR-RELS, A++ (Proposed), and the DNN-based HSCNN-R. Best results are shown in bold and underlined.

Approach	**Method**	**Mean Per-Image-Mean MRAE (%)**	**Mean Per-Image-99-Pt. MRAE (%)**
Pixel-based	A+	3.81	15.51
	PR-RELS	1.88	7.04
	A++ (Proposed)	**<u>1.69</u>**	7.78
DNN	HSCNN-R	1.73	**<u>6.53</u>**

Table 5.5: The reference number of model parameters, training time and testing (reconstruction) time of A+, PR-RELS, A++ (Proposed), and HSCNN-R.

Method	**Number of Parameters**	**Training time**	**Testing time (per image)**
A+	9.5×10^4	26.9 min	13.7 s
PR-RELS	2.6×10^3	42.2 min	5.9 s
A++ (Proposed)	7.6×10^5	3.4 h	1.1 min
HSCNN-R	1.7×10^7	35.5 h (with GPU)	13.0 s (with GPU)

cross validation process as detailed in Section 3.1.3. The A+, PR-RELS and the proposed A++ methods are pixel-based methods (i.e., the RGB-to-spectrum mapping operates at the pixel level), whereas HSCNN-R regresses 50 × 50 RGB image patches as a whole. A comparison of these methods' number of parameters used and time consumption for training and testing are given in Table 5.5.

Let us first look at the mean results. With A++, we reach even better mean performance than HSCNN-R which scores top in the NTIRE 2018 challenge [7]. Significantly, since A++ is pixel-based, this result teaches us that the patch-based mapping adopted by HSCNN-R may not bring in much useful information *on average*.

However, we do see HSCNN-R provides better worst-case performance (shown as the 99 percentiles) than the pixel-based methods. This indicates the possibility that DNNs might actually use the patch-based information to bound the performance of worst-case pixels. Still, the improvement is arguably small compared to PR-RELS (which is pixel-based).

Looking at the model complexity and time consumption evaluation (Table 5.5), we see that A++ is composed of more parameters than A+ and PR-RELS, but still being around 1 to 2 orders of magnitude less complex than the DNN methods. Nonetheless, in terms of testing time, our current implementation of A++ takes much longer than other compared algorithms. We note that our A++ implementation does not involve any code optimisation, e.g., parallel programming, and therefore a better time performance should be possible.

5.5 Discussion

In this chapter, we challenged ourselves to surpass the leading DNN-based SR using only a pixel-based mapping model. We developed a new sparse coding architecture, called "A++", where an RGB is mapped to spectrum, firstly by a polynomial regression SR, and secondly by a linear SR map depending on the location of its first estimation in the spectral space. We show that this A++ method—despite being much simpler than the leading DNNs—provides leading SR performance.

Although as per our research interest (to see whether patch information is needed for top-performing SR) we design A++ to be a pixel-based method, a pixel-based mapping fundamentally cannot distinguish materials of the same RGB (since the same RGB will always map to the same spectral estimate). This limitation goes against the premise that hyperspectral imaging can distinguish materials that are not distinguishable by an RGB camera. Hence, for applications where this ability is crucial, A++ and all other pixel-based methods may not be competent. But, they still serve as a baseline to see if the patch-based DNNs indeed perform better in this regard.

Even though we are presenting a pixel-based algorithm, what we want to show here is that top DNNs do not perform better than the best pixel-based methods, and this calls into doubt the extent to which these algorithms can map the same RGB to different spectra depending on context. This does not mean we do not recognise the DNNs' premise—that materials and/or objects are identified deep in the network—is good. Unfortunately that premise is not delivered upon in the architectures that are currently used.

Chapter 6

Exposure Invariant Spectral Reconstruction

In the two previous chapters, we dedicated ourselves to advancing the performance of pixel-based SR methods. Here, we wish to investigate a general problem of "exposure invariance" in spectral reconstruction.

6.1 Introduction

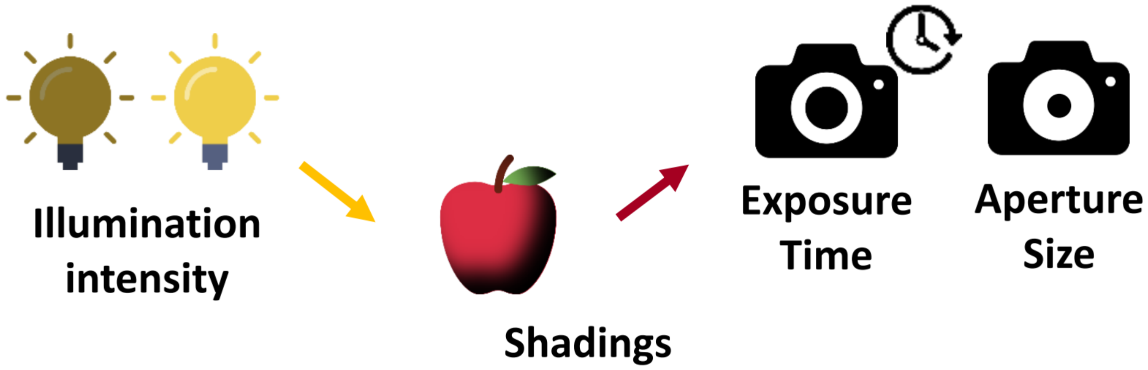


Figure 6.1: Sources of exposure change.

The spectral signal arriving at the camera sensor scales with exposure. As illustrated in Figure 6.1, the change of exposure can be caused by various factors, including when the user changes the camera's exposure time and/or aperture size settings, the prevailing illumination's brightness changes, and also the same object viewed in different parts of an image (so they are exposed to the illumination differently).

Let us denote $k > 0$ as a constant exposure scaling factor. In the event of exposure change, a radiance spectrum $\underline{\boldsymbol{h}}$ becomes $k\underline{\boldsymbol{h}}$. According to the hyperspectral-to-raw-RGB image formation (Section 2.1.3; Equation (2.4)), we see that the RGB generated from $k\underline{\boldsymbol{h}}$ will also be scaled by k:

$$\mathbf{R}^{\mathsf{T}}\underline{\boldsymbol{h}} = \underline{\boldsymbol{c}} \quad \Longleftrightarrow \quad \mathbf{R}^{\mathsf{T}}(k\underline{\boldsymbol{h}}) = k\underline{\boldsymbol{c}} \ . \tag{6.1}$$

Here, $\underline{\boldsymbol{h}}$ is the original (before scaling) ground-truth radiance spectrum, and $\underline{\boldsymbol{c}}$ is the original resulting RGB. The columns of $\mathbf{R}$ are the underlying camera's three discretised sensor spectral sensitivity functions (e.g., the CIE colour matching functions).

According to Equation (6.1), if $\underline{\boldsymbol{h}}$ and $\underline{\boldsymbol{c}}$ form a ground-truth pair, $k\underline{\boldsymbol{h}}$ and $k\underline{\boldsymbol{c}}$ can also be viewed as a ground-truth pair. Indeed, $k\underline{\boldsymbol{h}}$ and $k\underline{\boldsymbol{c}}$ might just be $\underline{\boldsymbol{h}}$ and $\underline{\boldsymbol{c}}$ viewed in a different exposure condition.

Now, in a hyperspectral image database (on which the SR algorithms are trained), there might already exist the same ground-truth $\underline{\boldsymbol{h}}$ and $\underline{\boldsymbol{c}}$ pairs scaled by many different k's. Nevertheless, this variation is often bounded—as usual the (hyperspectral) images are captured while avoiding scenes prevailed by under- and/or over-exposed pixels. Of course, it is important to preserve the visibility of the majority of the image content, so that the amount of useful information in an image can be maximised. And yet, in practice, problematic images do exist.

Another exposure-related problem is that we train the SR algorithms on *example* images. For example, an "apple" object may only appear once in one of the training images under a certain exposure condition, while in general an apple might appear brighter or darker, which will not be covered by the training dataset.

In regard to this issue, we wish to re-examine the existing SR algorithms on their capability to work on the original testing images scaled by different k factors, i.e., whether they are "exposure invariant". As shown in Figure 6.2, we found that the simple linear regression method [42] is able to return matching performance when the testing RGB image is at

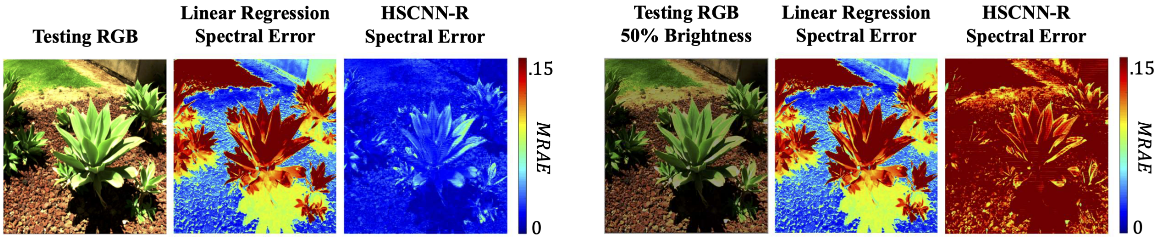

Testing RGB
Linear Regression Spectral Error
HSCNN-R Spectral Error
.15
MRAE
0
Testing RGB 50% Brightness
Linear Regression Spectral Error
HSCNN-R Spectral Error
.15
MRAE
0

Figure 6.2: Spectral reconstruction under varying exposure by linear regression and HSCNN-R. The spectral errors are calculated in MRAE.

50% brightness. In contrast, we see that the DNN-based HSCNN-R method significantly degraded when recovering the darker testing image.

Encouraged by this result, we further investigate the reason why some SR algorithms can preserve exposure invariance while others cannot. Based on this knowledge, we propose a new non-linear regression for SR by translating a technique used to solve colour correction in an exposure invariant way [28]. This technique is called Root-Polynomial Regression (RPR).

Additionally, since we found that most of the non-linear regressions and DNNs are not exposure invariant, we propose two approaches to imposing exposure invariance upon them. The first approach, called "chromaticity mapping", considers to separate the chromaticities of a pixel from its brightness scale, while only involve chromaticities in SR mapping. The second approach is "data augmentation", which is commonly adopted in learning algorithms to increase the model's generalisability. In data augmentation, we scale each training input (image patches or pixels) by a different randomly-selected k factor in training.

These two approaches are shown to be effective in terms of enforcing exposure invariance to an SR method. However, the overall performance of the models degrades from their original performance, likely due to the increase of data variation for the data augmentation approach, and the decrease of input information in the case of chromaticity mapping.

6.2 Observing the Existing SR Methods

Abstractly, the spectral reconstruction problem requires us to find a mapping function, denoted as $\Psi()$, that maps the camera responses $\underline{\boldsymbol{c}}_i$ to their corresponding spectra $\underline{\boldsymbol{h}}_i$:

$$\Psi(\underline{\boldsymbol{c}}_i) \approx \underline{\boldsymbol{h}}_i \ \ ; \ \ \forall i \ . \tag{6.2}$$

We say a method is exposure invariant if and only if:

$$\Psi(k\underline{\boldsymbol{c}}_i) = k\Psi(\underline{\boldsymbol{c}}_i) \approx k\underline{\boldsymbol{h}}_i \ ; \quad \forall k > 0 \ , \quad \forall i \ . \tag{6.3}$$

Let us consider the three SR approaches we have been evaluating in the past chapters: regression, A+ sparse coding, and Deep Neural Network (DNN).

6.2.1 Exposure Invariant Methods

First, in Linear Regression (LR), we have $\Psi(\underline{\boldsymbol{c}}_i) = \mathbf{M}\underline{\boldsymbol{c}}_i \approx \underline{\boldsymbol{h}}_i$ (Section 2.3.1; Equation (2.25)), which is clearly exposure invariant, i.e., $\Psi(k\underline{\boldsymbol{c}}_i) = k\Psi(\underline{\boldsymbol{c}}_i)$.

Then, in the original A+ sparse coding [1], locally in each neighbourhood the exposure-invariant linear regression mapping is used (Section 2.4.2; Equation (2.44)). That is, whether or not A+ is exposure invariant depends solely on whether scaling an input RGB will lead us to the same closest RGB dictionary member ($\mathbf{D}^c$ in Equation (2.37)) that defines its locality. Indeed, recall that we use the Prox^1 function (Equation (2.42)) which first *normalises* both the input query RGB $\underline{\boldsymbol{c}}_i$ and the RGB dictionary members in $\mathbf{D}^c$ before finding the closest RGB dictionary member. This normalisation ensures that the outcome of this process will not be influenced by exposure scaling. Hence, overall, A+ is an exposure invariant SR method.

6.2.2 Exposure Non-Invariant Methods

Regressions with a non-linear mapping function, including Polynomial Regression (PR) and Radial-Basis-Function Network (RBFN), follows $\Psi(\underline{\boldsymbol{c}}_i) = \mathbf{M}\varphi(\underline{\boldsymbol{c}}_i) \approx \underline{\boldsymbol{h}}_i$. Clearly, what determines the exposure invariance of both methods is whether $\varphi()$ preserves scale invariance.

For PR, polynomial expansion is used (Section 2.3.1; Equation (2.27)). Let us take the 2nd-order PR as an example (with $\underline{\boldsymbol{c}}_i = [R, G, B]^\intercal$), we get:

$$\begin{aligned} &\varphi(\underline{\boldsymbol{c}}_i) = [R, G, B, R^2, B^2, G^2, RG, GB, BR]^\intercal \\ \Longrightarrow\quad &\varphi(k\underline{\boldsymbol{c}}_i) = [kR, kG, kB, k^2R^2, k^2G^2, k^2B^2, k^2RG, k^2GB, k^2BR]^\intercal \ . \end{aligned} \tag{6.4}$$

Evidently, for PR, $\varphi(k\underline{\boldsymbol{c}}_i) \neq k\varphi(\underline{\boldsymbol{c}}_i)$, and thus $\Psi(k\underline{\boldsymbol{c}}_i) \neq k\Psi(\underline{\boldsymbol{c}}_i)$.

As for RBFN, each value in $\varphi(\underline{\boldsymbol{c}}_i)$ is a function of $\underline{\boldsymbol{c}}_i$'s ℓ_2 distance from a different preset RGB cluster centres (collectively these cluster centres are from the K-means clustering results; see Section 2.4.1). Here, unlike A+, the distances are calculated without normalising the input RGBs and the cluster centres. If we consider the jth centre in Equation (2.34), $\underline{\boldsymbol{c}}^j$, it is clear that $||k\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^j|| \neq ||\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{c}}^j||$. Therefore, collectively in RBFN we also get $\varphi(k\underline{\boldsymbol{c}}_i) \neq k\varphi(\underline{\boldsymbol{c}}_i)$, and then $\Psi(k\underline{\boldsymbol{c}}_i) \neq k\Psi(\underline{\boldsymbol{c}}_i)$.

Finally, the DNN solutions to spectral reconstruction, similar to the non-linear regressions, are also not exposure invariant. Recall the structure of a single neuron (Section 2.5; Equation (2.45)):

$$g(\underline{\boldsymbol{w}}^\intercal \underline{\boldsymbol{a}} + b) = a' \ , \tag{6.5}$$

where $\underline{\boldsymbol{a}}$, a', $\underline{\boldsymbol{w}}$ and b are the inputs, output, weights and bias, and $g()$ is the activation function. The offset term, b, alone indicates the neuron will not scale with exposure, i.e., if the input is $k\underline{\boldsymbol{a}}$ the output will not be ka'. Even when $b = 0$, the use of common non-linear activation functions leads to that the neuron output does not scale with the magnitude of the input. Note that the ReLU function [79] is, in fact, scale-invariant when $b = 0$, but its power of including non-linearity to the network highly depends on the non-zero bias terms. Given this view at the level of a single neuron, we can expect that DNNs—of whatever architectures—can hardly be exposure invariant by construction.

6.2.3 Comments on RELS, RELAD and A++

In Chapter 4, we proposed the RELS and RELAD formulations for the regressions. These two formulations alter the minimisation loss metric in training, but does not actually change how the regression mapping works: similarly to the conventional regression methods, a linear transformation matrix is used to map the input RGB $\underline{\boldsymbol{c}}_i$ (or $\varphi(\underline{\boldsymbol{c}}_i)$ for non-linear regressions) to its spectral estimate, but only the parameters in the linear matrix are optimised differently. Hence, whether or not a regression method is exposure invariant does not depend on whether we switch to use RELS or RELAD in training.

The A++ introduced in Chapter 5, as it stands, incorporates PR-RELS as its primary SR algorithm (see Section 5.3.1). As PR is not exposure invariant, a scaled input RGB might potentially be mapped to different cluster and subject to using different local linear regression in inference. Therefore, A++ is not exposure invariant if PR-RELS is used as the primary SR algorithm. (We note that if an exposure invariant SR is used as the primary SR algorithm of A++, it will become exposure invariant.)

6.3 Root-Polynomial Regression

From the analysis above, we see that non-linear mapping is usually the cause which makes an SR method not exposure invariant. Nevertheless, regarding the performances, non-linear methods are almost always better than linear mapping methods (e.g., LR and A+)—see in the baseline testing (Section 3.1.6; Table 3.3). To further improve the performance of linear regression while retaining its exposure-invariant property, we introduce a new non-linear fitting model for spectral reconstruction: Root-Polynomial Regression (RPR).

The proposed method is an extension from the work of Finlayson et al. [28] on root-polynomial regression for colour correction. While the colour correction problem considers the mapping from the camera-dependent RGB responses to the standard CIEXYZ colour space (or the display's RGB space), the spectral reconstruction problem seeks to estimate radiance spectra from the camera RGB.

Similarly to Polynomial Regression (PR), in RPR we expand each input RGB vector $\underline{\boldsymbol{c}}_i = [R, G, B]^\mathsf{T}$ to a non-linear polynomial series, only here the higher-order terms are compensated by the "root" operation of the same order. For example, the 2nd-, 3rd- and 4th-order root-polynomial expansions of $\underline{\boldsymbol{c}}_i$ are written as:

$$\begin{aligned}
\text{2nd-order } \varphi(\underline{\boldsymbol{c}}_i) &= \Big[R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{RB}\Big]^\mathsf{T} \\
\text{3rd-order } \varphi(\underline{\boldsymbol{c}}_i) &= \Big[R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{RB}, \\
&\quad \sqrt[3]{RG^2}, \sqrt[3]{GB^2}, \sqrt[3]{RB^2}, \sqrt[3]{GR^2}, \sqrt[3]{BG^2}, \sqrt[3]{BR^2}, \sqrt[3]{RGB}\Big]^\mathsf{T} \\
\text{4th-order } \varphi(\underline{\boldsymbol{c}}_i) &= \Big[R, G, B, \sqrt{RG}, \sqrt{GB}, \sqrt{RB}, \\
&\quad \sqrt[3]{RG^2}, \sqrt[3]{GB^2}, \sqrt[3]{RB^2}, \sqrt[3]{GR^2}, \sqrt[3]{BG^2}, \sqrt[3]{BR^2}, \sqrt[3]{RGB}, \\
&\quad \sqrt[4]{R^3G}, \sqrt[4]{R^3B}, \sqrt[4]{G^3R}, \sqrt[4]{G^3B}, \sqrt[4]{B^3R}, \sqrt[4]{B^3G}, \\
&\quad \sqrt[4]{R^2GB}, \sqrt[4]{G^2RB}, \sqrt[4]{B^2RG}\Big]^\mathsf{T} .
\end{aligned} \tag{6.6}$$

It is clear that $\varphi(k\underline{\boldsymbol{c}}_i) = k\varphi(\underline{\boldsymbol{c}}_i)$ always holds for root-polynomial expansion. Then, analogous to all other regression-based models, the ground-truth spectrum $\underline{\boldsymbol{h}}_i$ is estimated via a learnable linear transform: $\Psi(\underline{\boldsymbol{c}}_i) = \mathbf{M}\varphi(\underline{\boldsymbol{c}}_i) \approx \underline{\boldsymbol{h}}_i$. According to Chapter 4, this $\mathbf{M}$ matrix can be learned by minimising the conventional or per-channel least-squares criterion, or using the RELS or RELAD minimisation approach. Combined, we see that for RPR, $\Psi(k\underline{\boldsymbol{c}}_i) = \mathbf{M}\varphi(k\underline{\boldsymbol{c}}_i) = k\mathbf{M}\varphi(\underline{\boldsymbol{c}}_i) = k\Psi(\underline{\boldsymbol{c}}_i)$, $\forall i$. That is RPR is exposure invariant.

6.4 Chromaticity Mapping

In RPR, we formulate a non-linear mapping function that is by-construction invariant to the exposure change occurring in RGB captures. Here, we reformulate SR such that spectra are recovered from the *chromaticity*, which is a brightness-invariant feature of the RGBs.

6.4.1 Formulation

Every input RGB can be separated into a brightness scale times chromaticity:

$$\underline{\boldsymbol{c}}_i = ||\underline{\boldsymbol{c}}_i||_1 \cdot \left(\frac{\underline{\boldsymbol{c}}_i}{||\underline{\boldsymbol{c}}_i||_1} \right) \ ; \quad \forall i \ , \tag{6.7}$$

where $||\underline{\boldsymbol{c}}_i||_1 = |R| + |G| + |B|$ is equivalent to the component-sum of $\underline{\boldsymbol{c}}_i$, and $\frac{\underline{\boldsymbol{c}}_i}{||\underline{\boldsymbol{c}}_i||_1}$ defines the chromaticity of $\underline{\boldsymbol{c}}_i$.

With this separation, we apply the original SR algorithm $\Psi()$ (which might be exposure non-invariant) only to the chromaticity component to estimate spectrum $\underline{\boldsymbol{h}}_i$:

$$||\underline{\boldsymbol{c}}_i||_1 \cdot \Psi\left(\frac{\underline{\boldsymbol{c}}_i}{||\underline{\boldsymbol{c}}_i||_1} \right) \approx \underline{\boldsymbol{h}}_i \ ; \quad \forall i \ . \tag{6.8}$$

We call this SR approach *chromaticity mapping.*

Clearly, the outcome of Equation (6.8) is exposure invariant. Let us try inputting the same input $\underline{\boldsymbol{c}}_i$ scaled by k:

$$||k\underline{\boldsymbol{c}}_i||_1 \cdot \Psi\left(\frac{k\underline{\boldsymbol{c}}_i}{||k\underline{\boldsymbol{c}}_i||_1} \right) = k||\underline{\boldsymbol{c}}_i||_1 \cdot \Psi\left(\frac{\underline{\boldsymbol{c}}_i}{||\underline{\boldsymbol{c}}_i||_1} \right) \approx k\underline{\boldsymbol{h}}_i \ ; \quad \forall i \ , \tag{6.9}$$

which satisfies the condition of exposure invariance set out in Equation (6.3), irrespective of which $\Psi()$ function is in use.

We note that our chromaticity mapping approach has similar effect as Stiebel and Merhof [82], but arguably simpler in how we define the invariant brightness scale at each pixel (in our case we take the ℓ_1 norm of the input RGB, while Stiebel and Merhof use the vector projection of the input RGB onto the RGB of the recovered spectrum).

6.4.2 Training and Reconstruction

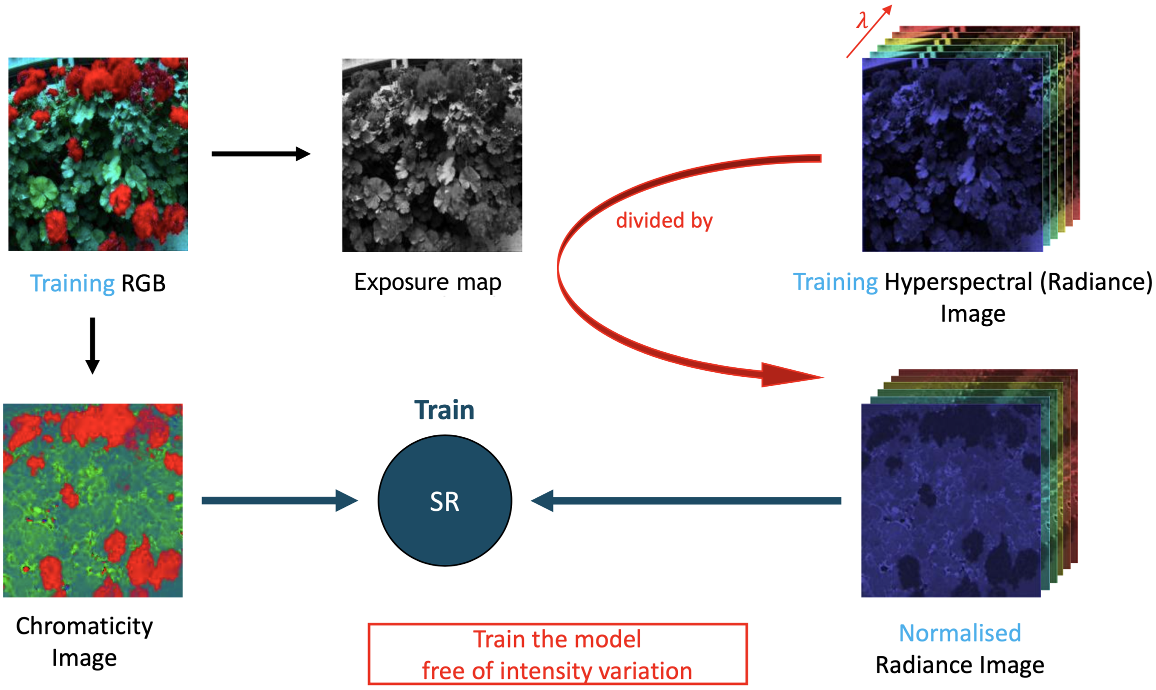

Training RGB
Exposure map
λ
divided by
Training Hyperspectral (Radiance) Image
Train
SR
Chromaticity Image
Train the model free of intensity variation
Normalised Radiance Image

Figure 6.3: The training scheme of the SR algorithm under chromaticity mapping.

Following the formulation in Equation (6.8), we divide the $||\underline{\boldsymbol{c}}_i||_1$ factor on both sides of the equation and derive the following training formulation for $\Psi()$:

$$\Psi\left(\frac{\underline{\boldsymbol{c}}_i}{||\underline{\boldsymbol{c}}_i||_1}\right) \approx \frac{\underline{\boldsymbol{h}}_i}{||\underline{\boldsymbol{c}}_i||_1} \ . \tag{6.10}$$

Extending to the image level, at each pixel we can calculate its $||\underline{\boldsymbol{c}}_i||_1$ and $\frac{\underline{\boldsymbol{c}}_i}{||\underline{\boldsymbol{c}}_i||_1}$ values. We call the $||\underline{\boldsymbol{c}}_i||_1$ components of all pixels the *exposure map* of the image. On the other hand, the chromaticity components of all image pixels, i.e., $\frac{\underline{\boldsymbol{c}}_i}{||\underline{\boldsymbol{c}}_i||_1}$, form the *chromaticity image*. With this image-level view, Equation (6.10) means to train the SR algorithm to map the chromaticity image to the ground-truth hyperspectral image pixel-wise divided by the exposure map, as illustrated in Figure 6.3. In inference, the algorithm-recovered *normalised* radiance image is going to (pixel-wise) multiply the exposure map to deliver the final estimate of the hyperspectral image.

This image-level description of the process is mainly to ease the understanding of applying chromaticity mapping in the context of DNN—where the algorithm maps image patches instead of pixel-RGBs. As we see in Figure 6.3, the chromaticity image loses the "intensity textures" of the original image, which could be a potential cue for the DNNs to recover spectra.

To retain this intensity cue in the training process **and** meanwhile ensuring the exposure invariance of the algorithms, we consider the *data augmentation* approach, introduced in the next section.

6.5 Data Augmentation

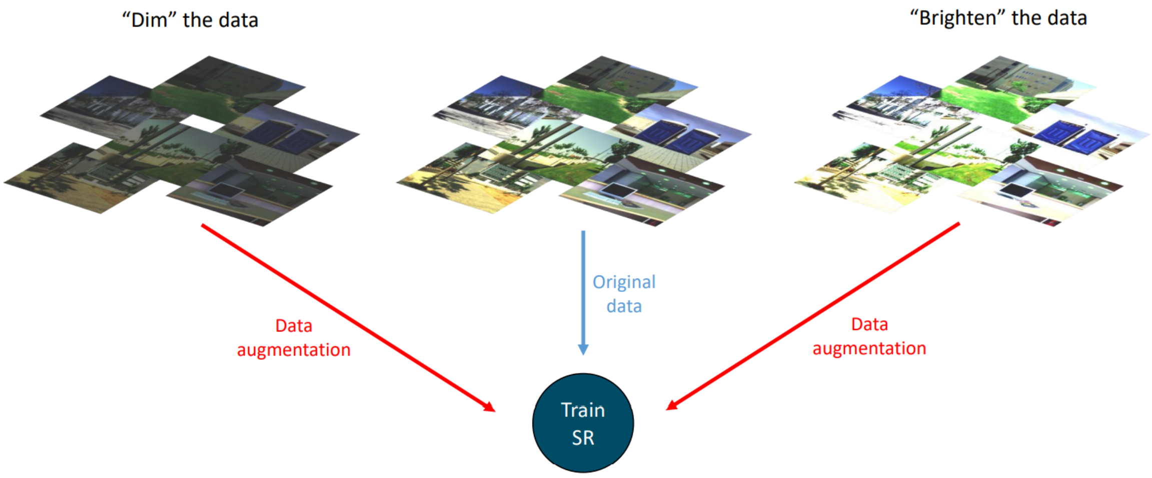


Figure 6.4: The training scheme of the SR algorithm under data augmentation.

Data augmentation refers to creating new training data by sensibly perturb the original data. As shown in Equation 6.3, applying any exposure scaling to the image formation process results in scaling up or down the RGB camera sensor response with the same factor. Therefore, we can add this intensity variation ourselves to the training images by scaling each pair of matching radiance spectra and RGB camera responses by a factor (either to dim or to brighten the data), as illustrated in Figure 6.4.

This way of adding a new dimension of variation effectively forces the machine learning model to fit, in addition to the original data, also all these new derived cases along this brightness dimension. However, it is not possible to cover all possible brightness variations. On one hand, intensity variation is *continuous*, while practically we can only select a finite number of different intensity scaling factors for augmentation. On the other hand, we do not know the actual bounds within which the object colours in a scene can vary. In practice, we can only decide on a range of intensity perturbation and a probability density function for randomly selecting scaling factors within this range.

As for a sensible choice for the probability density function, we point out a fact that the sequence of exposure adjustments in most of the cameras—both for the exposure time (shutter speed) and for aperture size—follow a *geometric progression*. More precisely, the available aperture sizes normally follow sequential scaling changes by $\sqrt{2}$, and the exposure time is adjusted by a factor of 2 between adjacent modes. This observation leads us to setting the probability density function as a uniform distribution on a *log* scale:

$$\log_\beta k \sim Uniform(-1, 1) \ , \tag{6.11}$$

where k is the randomly chosen scaling factor for data augmentation, and β decides the range of variation of k to be $[\frac{1}{\beta}, \beta]$.

In Figure 6.5, we compare the proposed distribution ($\beta = 10$; right panel of Figure 6.5) with the straightforward uniform distribution between $[0, 10]$ (left panel). From both distributions we drew 5000 random numbers and show the histogram with 100 bins on the log scale (linear to the geometric progression of the exposure modes). Evidently, the straightforward uniform distribution generates exponentially more bright scaling factors than the dim ones,

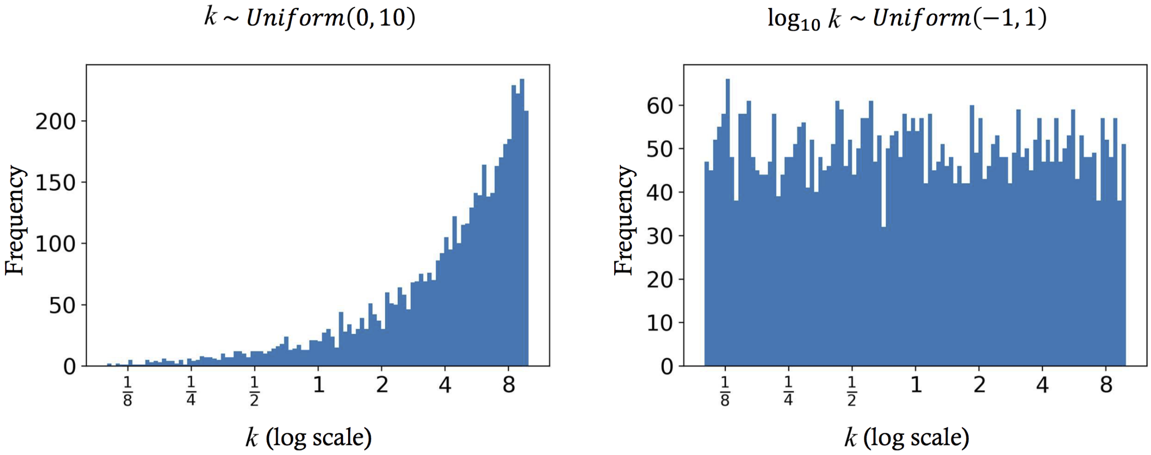


Figure 6.5: The comparison between drawing the scaling factor k from the straightforward uniform probability distribution (**left**) and from our proposed distribution (**right**).

while our proposed distribution suggests equal chances for bright and dim factors to be selected.

6.5.1 Training Setups for Regression and DNN

Most regression models we have discussed so far are optimised in closed form, except for our proposed RELAD optimisation approach (Chapter 4). In terms of closed-form optimisation, the minimisation is immediately decided given the training data. Hence, for the regressions' data augmentation we simply apply a random scaling factor on each pixel of the training images. These random factors are selected following the random distribution in Equation (6.11) with a fixed β factor which determines the range of the augmented brightness variation. We will present a test for using several different β's on the regressions' data augmentation in the experimental section (next section) of this chapter.

In contrast, DNN is optimised iteratively, i.e., the same training data passes through the DNN model multiple times while optimises the same set of model parameters. We can make use of this iterative process and apply different random scaling factors even for the same input data during different training iterations. In other words, we draw different random scaling factors from Equation (6.11) for different input training image patches **and** the same patches at different training iterations. Likewise, we will test for the data augmentation efficacy on the DNN with different β values used in the random distribution.

6.6 Experiments

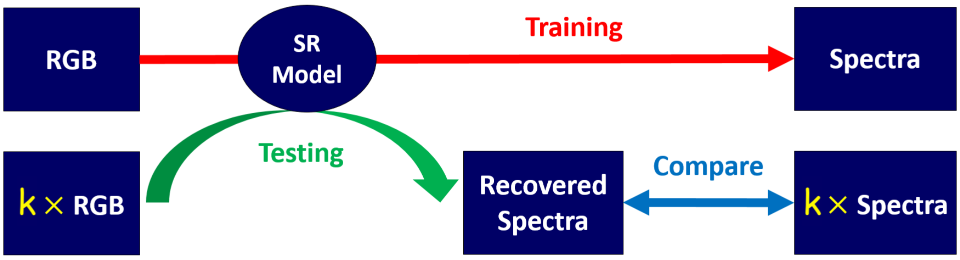


Figure 6.6: The flow chart of our exposure invariance testing scheme for SR.

Here, to test the exposure invariance of the trained SR methods (either trained originally, using chromaticity mapping, or data augmentation), we design a new evaluation scheme as shown in Figure 6.6. In effect, we multiply all testing RGB images by a factor of k, which leads to the target ground-truth hyperspectral images also scaled by k. We test $k = 0.5$ and $k = 2$, i.e., half or double the exposure of the original testing images. For comparison, We also present the result of $k = 1$, that is using the original testing images for testing.

Note that we manipulate the image's brightness scaling in *floating point numbers*, thus when scaling with $k < 1$ we will not lose the information in the dark image region due to digitisation, and we do not cap the overexposure level when $k > 1$ is considered. This is because the training and testing RGB images are simulated from the hyperspectral images, and there is no indication where the dynamic range of the underlying RGB camera actually is—the derived RGB values are at a relative intensity scale without any digitisation encoding.

The results we are going to present in this section are 4-fold cross-validated results following the CV procedure in Section 3.1.3.

Table 6.1: Cross-validated mean per-image-mean and per-image-99-percentile (Pt.) MRAE performance under testing exposure scaling $k = 1$, 0.5 and 2. Best results are shown in bold and underlined.

		Mean Per-Image-Mean MRAE (%)			**Mean Per-Image-99-Pt. MRAE (%)**		
	Method	**k = 1**	**k = 0.5**	**k = 2**	**k = 1**	**k = 0.5**	**k = 2**
Exposure Invariant	LR (LS)	6.24	6.24	6.24	16.95	16.95	16.95
	LR (RELS)	5.63	5.63	5.63	14.09	14.09	14.09
	LR (RELAD)	5.36	5.36	5.36	16.56	16.56	16.56
	A+ (LS)	3.81	3.81	3.81	15.52	15.52	15.52
	A+ (RELS)	3.60	3.60	3.60	13.50	13.50	13.50
	A+ (RELAD)	3.49	**3.49**	**3.49**	14.02	14.02	14.02
Exposure Invariant (Proposed)	RPR (LS)	4.69	4.69	4.69	16.97	16.97	16.97
	RPR (LS$^{\text{pc}}$)	4.31	4.31	4.31	14.98	14.98	14.98
	RPR (RELS)	4.18	4.18	4.18	12.94	**12.94**	**12.94**
	RPR (RELAD)	4.71	4.71	4.71	14.32	14.32	14.32
Exposure Non-Invariant	RBFN (LS)	2.06	18.58	8.74	7.89	41.94	24.14
	RBFN (RELS)	1.98	15.76	10.01	7.40	28.62	31.42
	PR (LS)	1.95	9.60	13.04	7.10	16.64	120.69
	PR (RELS)	1.88	10.97	21.60	7.04	18.74	248.48
	HSCNN-R	1.73	16.41	6.39	**6.53**	35.07	14.15
	A++	**1.69**	8.05	6.87	7.78	17.31	21.81

6.6.1 Existing Methods and RPR

Let us first test the exposure invariance of the regressions, A+ sparse coding, and DNN methods we have considered or proposed in previous chapters (e.g., regressions with RELS and/or RELAD minimisation and A++). Also the proposed RPR regression method optimised with the LS, LS$^{\text{pc}}$, RELS and RELAD approaches are tested and compared here.

In Table 6.1 we see the mean and the worst-case performances under varying exposure conditions: $k = 1$, 0.5 and 2. Interestingly, as we throw exposure change into the mix, we end up with two types of methods. First are the exposure-invariant methods, including LR, A+ and our proposed RPR methods. These methods maintain the *exact* performance as we adjust the images to be darker or brighter. The other type is exposure non-invariant methods, including RBFN, PR, HSCNN-R, and A++. Those methods perform particularly better when tested on the original testing image set (i.e., $k = 1$), while significantly degrades under exposure change.

Now let us look at the DNN-based HSCNN-R as an example. For the original testing image HSCNN-R works well, with the second-lowest mean and the lowest worst-case errors. However, as we darken the image by 50% (i.e., $k = 0.5$), HSCNN-R performs far worse than the much simpler linear regression (LR) and A+ sparse coding method—its MRAE is more than double the figure for LR. With respect to a doubling of exposure ($k = 2$), we see that HSCNN-R does not perform as badly as under the half exposure, though it could only perform on par with LR in mean and with A+ in the worst-case scenario.

Next, in terms of mean performance, our exposure-invariant RPR method (in general for different minimisation approaches used) performs slightly worse than the sparse-coding-based A+, while its RELS variant provides top worst-case performance (per-image 99 percentile errors) overall at half and double testing exposure conditions.

6.6.2 Effectiveness of Chromaticity Mapping

Using chromaticity mapping makes those SR algorithms which were exposure non-invariant now exposure invariant. Hence, in Table 6.2 we simply show the exposure-invariant mean and worst-case results (the same performances are shared by all testing exposures, i.e., $k = 1$, 0.5 and 2). To ease the comparison, we also include the methods that are by-construction exposure invariant (verified in Table 6.1) in the first tier of Table 6.2 (we shrink the list of by-construction exposure invariant methods to only include the optimisation methods that suggest the best mean and/or 99-percentile performance for each regression algorithm).

Clearly, our chromaticity mapping method pushes forward the frontier of how an exposure-invariant SR can perform. Indeed, originally, the best mean exposure-invariant performance was delivered by A+ (RELAD) with 3.49% per-image-mean MRAE, and the best exposure-

Table 6.2: Exposure-invariant mean per-image-mean and per-image-99-percentile (pt) MRAE performance under testing exposure scaling $k = 1$, 0.5 and 2, by construction or using the chromaticity mapping technique. Best results are shown in bold and underlined.

	Method	**Exposure-Invariant Mean Per-Image-Mean MRAE (%)**	**Exposure-Invariant Mean Per-Image-99-Pt. MRAE (%)**
By-Construction Exposure Invariant	LR (RELS)	5.63	14.09
	LR (RELAD)	5.36	16.56
	RPR (RELS)	4.18	12.94
	A+ (RELS)	3.60	13.50
	A+ (RELAD)	3.49	14.02
Using Chromaticity Mapping (Proposed)	RBFN (LS)	3.88	13.97
	RBFN (RELS)	3.76	13.06
	PR (LS)	4.20	13.14
	PR (RELS)	4.06	**<u>12.64</u>**
	HSCNN-R	**<u>3.16</u>**	13.43
	A++	3.58	13.43

invariant per-image-99-percentile MRAE was reached by RPR (RELS) at 12.94%. With chromaticity mapping, the best-performing exposure invariant SR is now 3.16% for mean MRAE performance delivered by HSCNN-R, and 12.64% for 99-percentile performance by PR with RELS minimisation.

We also notice that the chromaticity mapping method trades off the exposure non-invariant methods' original testing performance for exposure invariance. Indeed, we see in Table 6.1 that the mean MRAE delivered by the exposure non-invariant methods under the original testing exposure (i.e., $k = 1$) ranges from 1.69% to 2.06%, while now with chromaticity mapping, their performances degrade to the range between 3.16% and 4.20%. We believe it is more important to make sure an SR algorithm is exposure invariant as opposed to besting the performance at a fixed exposure while failing to work when the exposure changes.

6.6.3 Effectiveness of Data Augmentation

Unlike chromaticity mapping, data augmentation does not ensure exposure invariance will always hold. Indeed, its range of effectiveness is limited by the range of the augmentation (controled by the β factor in Equation (6.11)), and whether the algorithms are able to learn how to *completely* remove the effect of exposure change is not promised, i.e., it is still possible that the algorithms' exposure non-invariance is only *mitigated* instead of being totally solved.

Following the evaluation workflow in Figure 6.6, we test the RBFN (LS), PR (LS) and HSCNN-R methods while adopted different β values in data augmentation. For RBFN and PR, we tried $\beta = 2.5$, 5, 7.5 and 10, whereas for HSCNN-R, we only tested $\beta = 5$ and 10. Note that $\beta \geq 2$ is required, such that the augmented range $[\frac{1}{\beta}, \beta]$ (see Equation (6.11)) covers the testing exposures $k = 0.5$ and $k = 2$.

The mean per-image-mean MRAE and per-image-99-percentile results are given in Table 6.3. The mean results are also visualised in Figure 6.7, where we also plot three horizontal dotted lines respectively represent the exposure-invariant performances of LR, RPR and A+. The "No Augmentation" header in Table 6.3 and "No Aug." in Figure 6.7 refers to the performance of the methods without data augmentation (quoted from Table 6.1). All regression methods shown here are optimised using the conventional LS minimisation.

First, we see that for all three tested methods, data augmentation indeed stabilises their cross-exposure-condition performance, in comparison to the significant degradation in the original case (where no data augmentation was adopted).

As expected, with data augmentation we do not get the exact performance match across testing exposures like using chromaticity mapping—especially if we see the regression-based RBFN and PR, their performance under half exposure ($k = 0.5$) appears to be worse than under the original ($k = 1$) and double exposure ($k = 2$). As for HSCNN-R, though it also does not suggest exact performance match under different exposure conditions, it is much more stable compared to the regressions. Indeed, the error differences between different tested exposures are much smaller.

Table 6.3: The dependency of mean per-image-mean and per-image-99-percentile MRAE spectral accuracy on the β factor (Equation (6.11)) used for data augmentation. All SR methods are tested under original ($k = 1$), half ($k = 0.5$) and double exposure settings ($k = 2$).

	Mean Per-Image-Mean MRAE (%)								
	No Augmentation			$\beta = \mathbf{2.5}$			$\beta = \mathbf{5}$		
Method	**k = 1**	**k = 0.5**	**k = 2**	**k = 1**	**k = 0.5**	**k = 2**	**k = 1**	**k = 0.5**	**k = 2**
RBFN (LS)	2.06	18.58	8.74	4.20	5.67	4.33	6.19	6.02	5.30
PR (LS)	1.95	9.60	13.04	3.50	5.01	3.57	4.72	5.40	3.80
HSCNN-R	1.73	16.41	6.39	-	-	-	2.91	2.92	2.81
				$\beta = \mathbf{7.5}$			$\beta = \mathbf{10}$		
Method				**k = 1**	**k = 0.5**	**k = 2**	**k = 1**	**k = 0.5**	**k = 2**
RBFN (LS)				6.82	7.05	6.40	7.37	7.75	6.98
PR (LS)				5.25	5.72	4.45	5.74	6.03	5.13
HSCNN-R				-	-	-	2.97	2.97	2.96

	Mean Per-Image-99-Percentile MRAE (%)								
	No Augmentation			$\beta = \mathbf{2.5}$			$\beta = \mathbf{5}$		
Method	**k = 1**	**k = 0.5**	**k = 2**	**k = 1**	**k = 0.5**	**k = 2**	**k = 1**	**k = 0.5**	**k = 2**
RBFN (LS)	7.89	41.94	24.14	11.50	13.25	12.20	16.35	16.32	15.38
PR (LS)	7.10	16.64	120.69	10.79	12.36	11.59	12.96	13.80	12.05
HSCNN-R	6.53	35.07	14.15	-	-	-	10.49	10.53	10.04
				$\beta = \mathbf{7.5}$			$\beta = \mathbf{10}$		
Method				**k = 1**	**k = 0.5**	**k = 2**	**k = 1**	**k = 0.5**	**k = 2**
RBFN (LS)				18.16	18.30	18.10	19.97	19.91	20.25
PR (LS)				13.82	14.60	12.68	14.68	15.39	13.41
HSCNN-R				-	-	-	11.05	11.05	11.06

We see that for all RBFN, PR and HSCNN-R, the overall mean and 99-percentile performances degrade to higher error levels compared to when they are trained without data augmentation and tested under the original exposure condition. This trend is similar to what we have observed in the chromaticity mapping results. Nevertheless, in the case of applying data augmentation on RBFN and PR, the overall error levels rise as a larger β is used in data augmentation. In contrast, the β dependence is much smaller for HSCNN-R. This means the increase of the range of exposure variation in training data poses as a challenge to the regression methods, while less so in the case of using a DNN. We also see the degraded performance levels of RBFN and PR are already worse than the by-construction exposure invariant methods such as A+, RPR, or even LR, while the data-augmented HSCNN-R still holds some advantage against those methods.

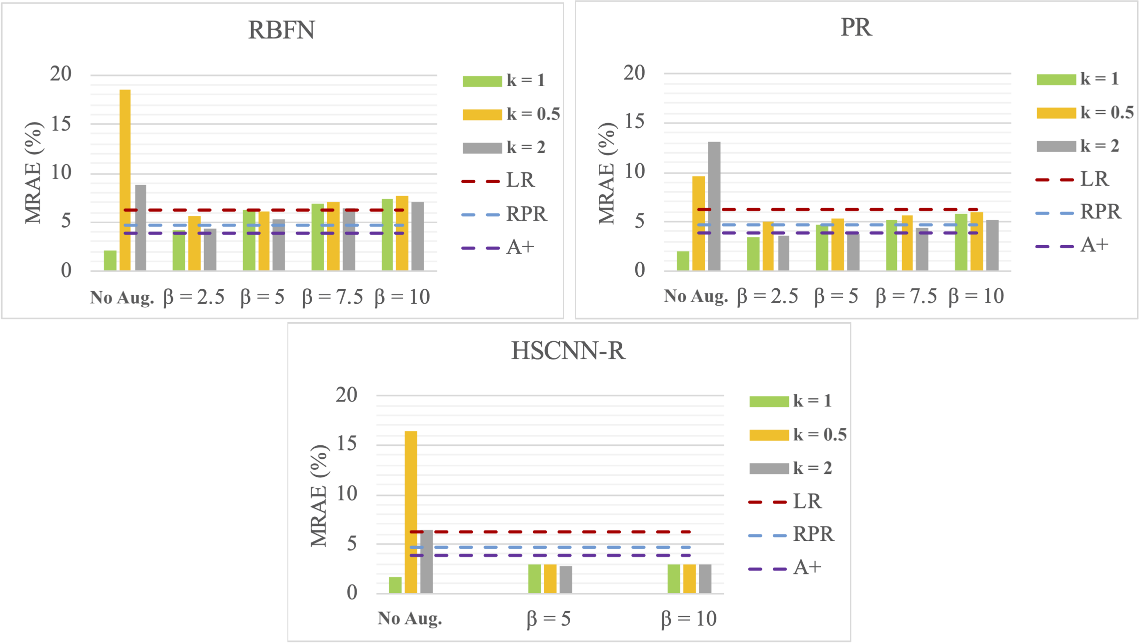


Figure 6.7: Visualising the performance and generalisability (in Mean MRAE) with respect to different β factors chosen.

As the results suggest, data augmentation is more suitable for the DNN-based HSCNN-R compared to the regression-based RBFN and PR. We can also compare the results of using data augmentation and chromaticitiy mapping on HSCNN-R. Clearly, if data augmentation's limited effective range of exposure condition generalisability is not of concern, the data-augmented HSCNN-R performs better than HSCNN-R with chromaticity mapping in both mean and 99-percentile MRAE results, delivering the best exposure invariant SR performance overall (compared to all by-construction exposure invariant methods and methods using chromaticity mapping).

6.7 Summary

In this chapter we reveal the problem of exposure invariance in SR, which refers to an SR algorithm's capability of working when the exposure condition in testing changes. We found that linear-mapping-based methods such as Linear Regression (LR) and A+ sparse coding are exposure invariant, while all non-linear-mapping-based methods—including all better performing SR algorithms in previous benchmarks such as RBFN, PR, HSCNN-R and A++—are shown to be exposure non-invariant, with drastic performance degradation.

Given this finding, we first proposed a standalone new SR method, the Root-Polynomial Regression (RPR) method, which is by-construction exposure invariant while using a non-linear mapping function. With the boost from the advanced RELS minimisation for regressions we proposed in Chapter 4, RPR is shown to provide leading worst-case performance (mean per-image-99-percentile MRAE) compared to all by-construction exposure invariant methods.

Then, we propose two general modification frameworks—chromaticity mapping and data augmentation—an SR algorithm can adopt to ensure or improve its exposure invariance.

In chromaticity mapping, we remove the pixel brightness variation before training an SR algorithm, while later re-applying the pixel brightness back to the recovered brightness-normalised spectra. This approach ensures *perfect* exposure invariance. However, we note that the pixel brightness information excluded in training might provide useful information for recovering spectra.

Alternatively, we formulated a data augmentation framework, where we apply random exposure scaling factors to the training data so that the SR algorithms can learn the exposure variation presented in testing, save that the range of its generalisability is limited by the range of random exposure scaling factors applied. We see that for the regression-based RBFN and PR methods, the problem of their exposure non-invariance can only be *partially* solved via data augmentation—while mitigated, there still exists notable performance variation across different testing exposure conditions. In contrast, the DNN-based HSCNN-R can learn via data augmentation to balance its performance across different testing exposures. And, the data-augmented HSCNN-R was found to be the best exposure invariant SR method we have benchmarked here.

Chapter 7

Physically Plausible Spectral Reconstruction

This chapter begins with an observation of the SR problem. Specifically, we found that for all regression and DNN methods we considered in this thesis, as we reintegrated the recovered spectra with the camera spectral sensitivities, we arrived at RGBs that were different from the input RGBs from which the spectra were recovered. We call this the *physical plausibility* problem of SR. The methods developed in this chapter resolve this problem (and apply to any and all SR approaches).

7.1 Introduction

Clearly, spectra and RGBs are physically related: abstractly, an RGB camera weighted-sums the spectral signals coming from the scene following the spectrally-varying sensitivity (weighting) functions of three different colour sensors, resulting in the 3-value RGB camera responses. This relation is also known as the *colour image formation* formula, as introduced in Section 2.1.1; Equation (2.1), or Equation (2.4) as its discrete version. Yet, this physical fact is generally not employed by the best SR algorithms. It is shown in Arad et al. [8] that even top DNN approaches recover spectral estimates that do not physically reproduce the same input RGBs.

Apart from the theoretical inconsistency, physically implausible spectral reconstructions imply that these algorithms *alter* the original colours when converting them to the spectral

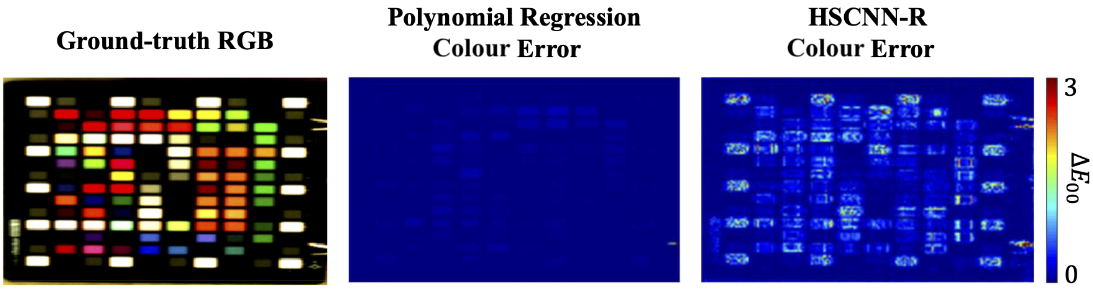


Figure 7.1: The colour errors introduced by polynomial regression SR [25] (**left**) and HSCNN-R [80] (**right**). The colour errors are measured in CIE ΔE 2000 (ΔE_{00}) [78].

form. Surely, this can be a serious problem in some practical applications where colour fidelity is of concern, e.g., art archiving and conservation [67]. In Figure 7.1, we give an example of the colour errors introduced by polynomial regression SR [25] and one of the leading deep-learning models, HSCNN-R [80]. We can clearly see that HSCNN-R—despite claiming the state-of-the-art spectral accuracy [7]—performs much worse in colour than the regression-based polynomial regression model. However, the very existence of the non-zero colour errors indicates that *both methods are physically implausible.*

Equally, aside from improved colour fidelity, if we can make SR physically plausible then we should be able to recover spectra with greater accuracy than has been achieved hitherto. Indeed, there are already studies that use the physics of image formation to improve SR. Based on a 3-D linear model (Section 2.2.1), Agahian et al. [2] proposed to characterise each 3-D reflectance dynamically while putting more weights on the reflectances of close-by colours. Zhao et al. [103] developed a matrix-R approach to colorimetrically post-facto correct the linear regression-based SR. Morovic and Finlayson [60] used metamer sets [29] as the physical constraints of Bayesian inference (which leads to a physically plausible method). Bianco [13] proposed an iterative algorithm which includes colour difference in the optimisation function. However, the performance of these methods—developed over 10 years ago—is unlikely to be competitive with today's leading methods.

We see that the physical prior is also incorporated in some of the top methods in the NTIRE 2020 Spectral Recovery Challenge [8]: the first-place winner Li et al. [53] included colour difference in their learning cost function, and Joslyn Fubara et al. [46] designed an unsupervised learning approach based on the physics prior. However, even these last two methods still recover spectra of wrong colours [8].

In this chapter we develop a generic reformulation of the SR problem and show that in so doing we can, in a universally applicable way, make all SR algorithms physically plausible. We propose two approaches. First, in the training data preparation stage. Given the ground-truth spectra, we separate them into two spectral components—fundamental metamer and metameric black. The former is unique for (and can be directly derived from) a given RGB, that is all spectra with the same RGB colour will have the same and unique fundamental metamer. In contrast, the non-unique metameric black is the component that distinguishes those same-coloured spectra. Given this insight we propose to train the SR algorithms to recover only the metameric black component of the spectra, while the fundamental metamers are derived directly from the RGBs and kept out of the learning process. As such, the physical plausibility is ensured for the trained algorithms, and we show that this method also does not deteriorate the performance of the original algorithms.

The second method we propose is a post-facto process resembling the matrix-R approach proposed by Zhao et al. [103]. In this approach we do not retrain the SR algorithms. Instead, we simply replace the (wrong) fundamental metamers of all reconstructed spectra by the correct ones derived from the input RGBs. Importantly, we will also formulate a mathematical proof which suggests that this post-facto step will only improve or retain the RMSE spectral errors for any existing SR methods.

7.2 Enforcing Physical Plausibility in Training

Conceptually, we wish to formulate the representation of a “plausible set” which is the set of spectra which, when captured by an RGB camera (Equation (2.4)), derive a given RGB. Then, in the context of SR, we say an SR algorithm is physically plausible if, given any input RGB, the algorithm always recover spectrum within its plausible set.

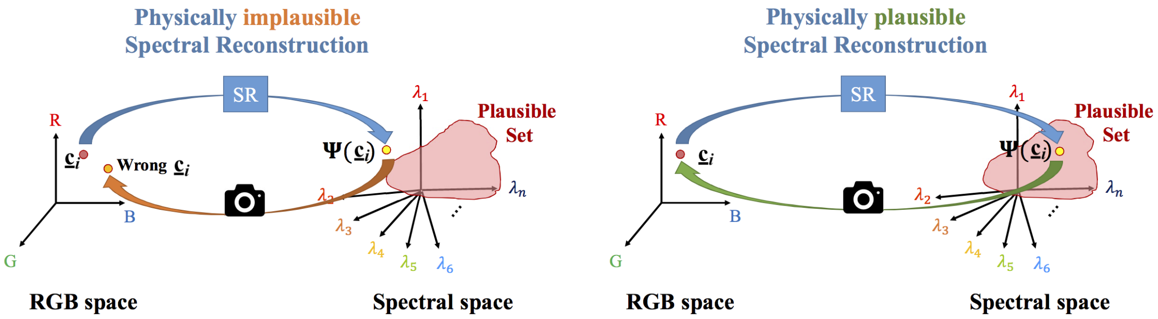


Figure 7.2: Physically implausible (**left**) and physically plausible spectral reconstruction (**right**).

Unfortunately, most of the current algorithms are physically *im*plausible. We contrasts SR algorithms of these two different qualities in Figure 7.2. On the left we show implausible spectral reconstruction, where an image RGB is mapped to a spectrum outside the plausible set. In this scenario, when the recovered spectrum is integrated with the camera sensors, the resulting RGB is different from the one we started with. On the right of Figure 7.2, we show physically plausible spectral reconstruction. Here the recovered spectrum is inside the plausible set and so integrates to the same RGB that we started with.

Mathematically, the colour image formation (Equation (2.4)) entails that given an $n \times 3$ camera spectral sensitivity matrix $\mathbf{R}$, all RGBs, $\underline{\boldsymbol{c}}_i$, are derived from their hyperspectral measurements, $\underline{\boldsymbol{h}}_i$, by:

$$\mathbf{R}^{\mathsf{T}}\underline{\boldsymbol{h}}_i = \underline{\boldsymbol{c}}_i \ ; \quad \forall i \ . \tag{7.1}$$

A spectral reconstruction algorithm is said to be physically plausible if and only if for all RGBs, the recovered spectra, $\Psi(\underline{\boldsymbol{c}}_i)$, when taking an inner product with $\mathbf{R}$, derive $\underline{\boldsymbol{c}}_i$ exactly:

$$\mathbf{R}^{\mathsf{T}}\Psi(\underline{\boldsymbol{c}}_i) = \mathbf{R}^{\mathsf{T}}\underline{\boldsymbol{h}}_i = \underline{\boldsymbol{c}}_i \ ; \quad \forall i \ . \tag{7.2}$$

7.2.1 The Plausible Set

Based on Equation (7.2), the plausible set of a given RGB, $\underline{\boldsymbol{c}}_i$, is defined as:

$$\mathcal{P}(\underline{\boldsymbol{c}}_i;\mathbf{R}) = \left\{\widehat{\underline{\boldsymbol{h}}}_i \;\middle|\; \mathbf{R}^\top\widehat{\underline{\boldsymbol{h}}}_i = \underline{\boldsymbol{c}}_i\right\} \; ; \quad \widehat{\underline{\boldsymbol{h}}}_i = \Psi(\underline{\boldsymbol{c}}_i) \; . \tag{7.3}$$

Here, $\widehat{\underline{\boldsymbol{h}}}_i \in \mathcal{P}(\underline{\boldsymbol{c}}_i;\mathbf{R})$ represents all spectra that integrate into $\underline{\boldsymbol{c}}_i$, and $\Psi(\underline{\boldsymbol{c}}_i) = \widehat{\underline{\boldsymbol{h}}}_i \in \mathcal{P}(\underline{\boldsymbol{c}}_i;\mathbf{R})$ represents a physically plausible spectral recovery. Here, $\mathcal{P}(\underline{\boldsymbol{c}}_i;\mathbf{R})$ defines a "candidate set" of $\underline{\boldsymbol{c}}_i$ for physically plausible SR. Of course, this candidate (plausible) set also includes the actual ground-truth $\underline{\boldsymbol{h}}_i$ (without the $\widehat{}$ symbol).

Clearly, this plausible set concept resembles the Metamer Sets [60] introduced in Section 2.2.3. As a recap, the metamer set $\mathcal{M}(\underline{\boldsymbol{c}}_i;\mathbf{\Lambda})$ defines a set of *reflectances* that have the same colour $\underline{\boldsymbol{c}}_i$ under the fixed illumination and camera condition defined by the lighting matrix $\mathbf{\Lambda}$ (see Equation (2.9)). The only differences between the two sets are that, first, in plausible sets we do not consider fixed illumination—the spatially and scene-wise varying illumination spectra are considered as part of the to-be-recovered spectral radiance information—and secondly, with radiance instead of reflectance considered, the colours are derived by applying $\mathbf{R}^\top$ (to the radiances) instead of $\mathbf{\Lambda}$ (which operates on reflectances).

With this analogy in place, we can derive a Metamer-Sets-like representation of Equation (7.3) following the process detailed in Section 2.2.3: from Equation Equation (2.19) to (2.23), while using $\widehat{\underline{\boldsymbol{h}}}_i$ and $\mathbf{R}^\top$ to replace the reflectance's basis coefficient vector $\underline{\boldsymbol{\alpha}}_i$ and the lighting matrix $\mathbf{\Lambda}$.

Starting with splitting $\widehat{\underline{\boldsymbol{h}}}_i$ into two components with respect to $\mathbf{R}$:

$$\widehat{\underline{\boldsymbol{h}}}_i = \mathbf{P}^f\widehat{\underline{\boldsymbol{h}}}_i + \mathbf{P}^b\widehat{\underline{\boldsymbol{h}}}_i \; , \tag{7.4}$$

where

$$\begin{cases} \mathbf{P}^f = \mathbf{R}(\mathbf{R}^\mathsf{T}\mathbf{R})^{-1}\mathbf{R}^\mathsf{T} \\ \mathbf{P}^b = \mathbf{I} - \mathbf{P}^f \end{cases} . \tag{7.5}$$

In the parlance of linear algebra, $\mathbf{P}^f$ is a projection matrix that projects $\underline{\widehat{\boldsymbol{h}}}_i$ onto the 3-dimensional subspace spanned by the 3 columns of $\mathbf{R}$, i.e., $\mathcal{C}(\mathbf{R})$, and $\mathbf{P}^b$ is the projection matrix that projects $\underline{\widehat{\boldsymbol{h}}}_i$ onto the $(n-3)$-dimensional "nullspace" of $\mathcal{C}(\mathbf{R})$. Together these two spaces span the whole n-dimensional spectral space.

Then, we refer the former term of Equation (7.4) to as the *fundamental metamer* of $\underline{\widehat{\boldsymbol{h}}}_i$, denoted as $\underline{\widehat{\boldsymbol{h}}}_i^f$, and the latter to as its *metameric black*, denoted as $\underline{\widehat{\boldsymbol{h}}}_i^b$. By expanding these two components in Equation (7.4) using Equation (7.5), we get:

$$\begin{cases} \underline{\widehat{\boldsymbol{h}}}_i^f = \mathbf{P}^f\underline{\widehat{\boldsymbol{h}}}_i = \mathbf{R}(\mathbf{R}^\mathsf{T}\mathbf{R})^{-1}[\mathbf{R}^\mathsf{T}\underline{\widehat{\boldsymbol{h}}}_i] \\ \underline{\widehat{\boldsymbol{h}}}_i^b = \mathbf{P}^b\underline{\widehat{\boldsymbol{h}}}_i = \mathbf{B}^b\underline{\widehat{\boldsymbol{\beta}}}_i^b \end{cases} , \tag{7.6}$$

where the columns of $\mathbf{B}^b$ are the $n-3$ null space bases derived from $\mathbf{P}^b$ as its $n-3$ linearly independent columns ($\mathbf{P}^b$ is rank $n-3$) [83], and $\underline{\widehat{\boldsymbol{\beta}}}_i^b$ is a coefficient vector with length $n-3$. We use the Gram–Schmidt orthogonalisation procedure [22] to derive $\mathbf{B}^b$ from $\mathbf{P}^b$, which, additionally, ensures that the columns of $\mathbf{B}^b$ form a set of "orthogonal bases" of the $n-3$ spectral subspace (i.e., ${\mathbf{B}^b}^\mathsf{T}\mathbf{B}^b = \mathbf{I}$ where $\mathbf{I}$ is the $(n-3) \times (n-3)$ identity matrix) [83; 22].

Note that the columns of $\mathbf{R}$ and $\mathbf{B}^b$ combine to form a basis of the n-dimensional spectral space, where

$$\begin{cases} \underline{\widehat{\boldsymbol{h}}}_i^f \in \mathcal{C}(\mathbf{R}) \\ \underline{\widehat{\boldsymbol{h}}}_i^b \in \mathcal{C}(\mathbf{B}^b) \end{cases} ; \quad \forall i . \tag{7.7}$$

And,

$$\mathbf{R}^\mathsf{T}\mathbf{B}^b = \mathbf{0} , \tag{7.8}$$

where $\mathbf{0}$ is an $3 \times (n-3)$ matrix of zeros, signifying that all columns of $\mathbf{R}$ are orthogonal to all columns of $\mathbf{B}^b$. These are important properties of $\mathcal{C}(\mathbf{R})$ and $\mathcal{C}(\mathbf{B}^b)$ that will be used later in Section 7.3.

Returning to the definition of plausible set in Equation (7.3), the constraint, $\mathbf{R}^\mathsf{T}\underline{\widehat{\boldsymbol{h}}}_i = \underline{\boldsymbol{c}}_i$ directly applies to the fundamental metamer $\underline{\widehat{\boldsymbol{h}}}_i^f$ in Equation (7.6), deriving:

$$\underline{\widehat{\boldsymbol{h}}}_i^f = \mathbf{R}(\mathbf{R}^\mathsf{T}\mathbf{R})^{-1}\underline{\boldsymbol{c}}_i = \underline{\boldsymbol{h}}_i^f \ , \tag{7.9}$$

which is a fixed vector given $\underline{\boldsymbol{c}}_i$. Since in $\mathcal{P}(\underline{\boldsymbol{c}}_i, \mathbf{R})$ $\underline{\boldsymbol{c}}_i$ is fixed, and the ground-truth $\underline{\boldsymbol{h}}_i$ is also a member of this set, we know that all $\underline{\widehat{\boldsymbol{h}}}_i$ in this set have the same fundamental metamer component as the one the ground-truth $\underline{\boldsymbol{h}}_i$ has, i.e., $\underline{\boldsymbol{h}}_i^f$.

On the other hand, the colour formation constraint does not influence the metameric black component, leaving $\underline{\widehat{\boldsymbol{\beta}}}_i^b$ to be $n-3$ "free variables" that define the variation of $\underline{\widehat{\boldsymbol{h}}}_i$ in $\mathcal{P}(\underline{\boldsymbol{c}}_i; \mathbf{R})$.

Combined, we arrive at our final representation of the plausible set:

$$\mathcal{P}(\underline{\boldsymbol{c}}_i; \mathbf{R}) = \left\{ \underline{\boldsymbol{h}}_i^f + \mathbf{B}^b \underline{\widehat{\boldsymbol{\beta}}}_i^b \;\middle|\; \underline{\widehat{\boldsymbol{\beta}}}_i^b \in \mathbb{R}^{n-3} \right\} \tag{7.10}$$

($\underline{\boldsymbol{h}}_i^f$ is fixed and defined as in Equation (7.9)).

In effect, if we constrain an SR algorithm to only predict the $\underline{\widehat{\boldsymbol{\beta}}}_i^b$ information from the input $\underline{\boldsymbol{c}}_i$ (treating both $\underline{\boldsymbol{h}}_i^f$ and $\mathbf{B}^b$ as known factors without needing for an estimation), we can enforce a physically plausible SR. That is we wish to map from the input $\underline{\boldsymbol{c}}_i$ to $\underline{\widehat{\boldsymbol{\beta}}}_i^b$.

7.2.2 Deriving the Ground Truths

Again, $\mathcal{P}(\underline{\boldsymbol{c}}_i; \mathbf{R})$ defines a set of *candidates* for physically plausible spectral recovery from input $\underline{\boldsymbol{c}}_i$, with the actual ground-truth being a particular one of them. Let us denote $\underline{\boldsymbol{\beta}}_i^b$ as the $n-3$ coefficient vector that uniquely represents the ground-truth $\underline{\boldsymbol{h}}_i$ (out of all possible $\widehat{\underline{\boldsymbol{\beta}}}_i^b$ within $\mathcal{P}(\underline{\boldsymbol{c}}_i;\mathbf{R})$). Then, we can calculate the $\underline{\boldsymbol{\beta}}_i^b$ from $\underline{\boldsymbol{h}}_i$. First, according to the plausible set format, we write $\underline{\boldsymbol{h}}_i$ as:

$$\underline{\boldsymbol{h}}_i = \underline{\boldsymbol{h}}_i^f + \mathbf{B}^b\underline{\boldsymbol{\beta}}_i^b \ . \tag{7.11}$$

Next, we multiply ${\mathbf{B}^b}^{\mathsf{T}}$ in the front of both sides of Equation (7.11):

$${\mathbf{B}^b}^{\mathsf{T}}\underline{\boldsymbol{h}}_i = {\mathbf{B}^b}^{\mathsf{T}}\underline{\boldsymbol{h}}_i^f + [{\mathbf{B}^b}^{\mathsf{T}}\mathbf{B}^b]\underline{\boldsymbol{\beta}}_i^b \ . \tag{7.12}$$

As $\underline{\boldsymbol{h}}_i^f$ lies in the column space of $\mathbf{R}$ which is orthogonal to the subspace spanned by the columns of $\mathbf{B}^b$ (i.e., the nullspace of the column space of $\mathbf{R}$), we get ${\mathbf{B}^b}^{\mathsf{T}}\underline{\boldsymbol{h}}_i^f = \underline{\mathbf{0}}$ where $\underline{\mathbf{0}}$ is an $(n-3)$-vector of zeros. Then, since the columns of $\mathbf{B}^b$ are orthogonal bases, we know that ${\mathbf{B}^b}^{\mathsf{T}}\mathbf{B}^b = \mathbf{I}$. We arrive at:

$$\underline{\boldsymbol{\beta}}_i^b = {\mathbf{B}^b}^{\mathsf{T}}\underline{\boldsymbol{h}}_i \ , \tag{7.13}$$

which effectively derives the ground-truth $\underline{\boldsymbol{\beta}}_i^b$ (i.e., the factor to be recovered in our physically plausible SR) from the ground-truth hyperspectral measurement $\underline{\boldsymbol{h}}_i$.

7.2.3 Recovering Spectra within the Plausible Sets

Conventionally, spectral reconstruction algorithms such as regressions and DNNs are formulated to pursue the minimisation of recovery error between $\Psi(\underline{\boldsymbol{c}}_i)$ and $\underline{\boldsymbol{h}}_i$, where $\Psi()$ is an SR algorithm, $\underline{\boldsymbol{c}}_i$ is an input RGB, and $\underline{\boldsymbol{h}}_i$ is the ground-truth spectrum. Let us split the ground-truth $\underline{\boldsymbol{h}}_i$ into the fundamental metamer and metameric black components we introduced in Section 7.2.1. We get:

$$\Psi(\underline{\boldsymbol{c}}_i) \approx \underline{\boldsymbol{h}}_i^f + \underline{\boldsymbol{h}}_i^b \ , \tag{7.14}$$

where, *a priori*, the approximation errors can occur on both $\underline{\boldsymbol{h}}_i^f$ and $\underline{\boldsymbol{h}}_i^b$. Specifically, any error introduced on $\underline{\boldsymbol{h}}_i^f$ makes it not exactly $\underline{\boldsymbol{h}}_i^f$, and consequently makes the spectral recovery outside of $\mathcal{P}(\underline{\boldsymbol{c}}_i;\mathbf{R})$, representing a physically implausible spectral recovery.

Therefore, the key to physically plausible SR is to ensure that all recovered spectra have exactly the same fundamental metamer component as the ground-truth ones. Since all the ground-truth fundamental metamers can be calculated from the input RGBs and the spectral sensitivity matrix (Equation (7.9)), we can direct the SR algorithm only to focus on estimating $\underline{\boldsymbol{\beta}}_i^b$ (Equation (7.13)) which uniquely singles out the ground-truth within the plausible set.

We formulate:

$$\Psi'(\underline{\boldsymbol{c}}_i) = \underline{\boldsymbol{h}}_i^f + \mathbf{B}^b\Psi(\underline{\boldsymbol{c}}_i) \ ; \quad \Psi(\underline{\boldsymbol{c}}_i) \approx \underline{\boldsymbol{\beta}}_i^b \ ; \quad \forall i \ , \tag{7.15}$$

where $\Psi()$ is the original SR algorithm repurposed to approximate the vector $\underline{\boldsymbol{\beta}}_i^b$, and $\Psi'()$ is the resulting physically plausible SR.

To sum up, a visualisation of our physically plausible method in comparison to the conventional physically implausible approach is given in Figure 7.3. In the standard approach (top flowchart) the training/estimation scheme directly maps the RGBs to spectra. Here, $\underline{\boldsymbol{h}}_i$ may not integrate to $\underline{\boldsymbol{c}}_i$ (the RGB from which it was recovered). In the physically plausible approach (bottom flowchart), the reconstruction is split into two streams. In the first stream the fundamental metamer—which is the only part that contributes to the RGB formation—is calculated directly from the input RGB. Then, the second stream seeks to find the best estimate for the metameric black. By construction the recovered spectrum (the sum of the fundamental metamer and the metameric black) must integrate to the same RGB.

7.2.4 Output Layer of the DNNs

Unlike hyperspectral measurements where all measured numbers must be positive, the ground-truth $\underline{\boldsymbol{\beta}}_i^b$ vectors can include negative numbers. For regression, the output space does not limit the range of the ground-truth values. That is, the same regression model that was used to recover hyperspectral measurements before, can now be used to recover the $\underline{\boldsymbol{\beta}}_i^b$ information directly (without any additional adaption). For HSCNN-R, however,

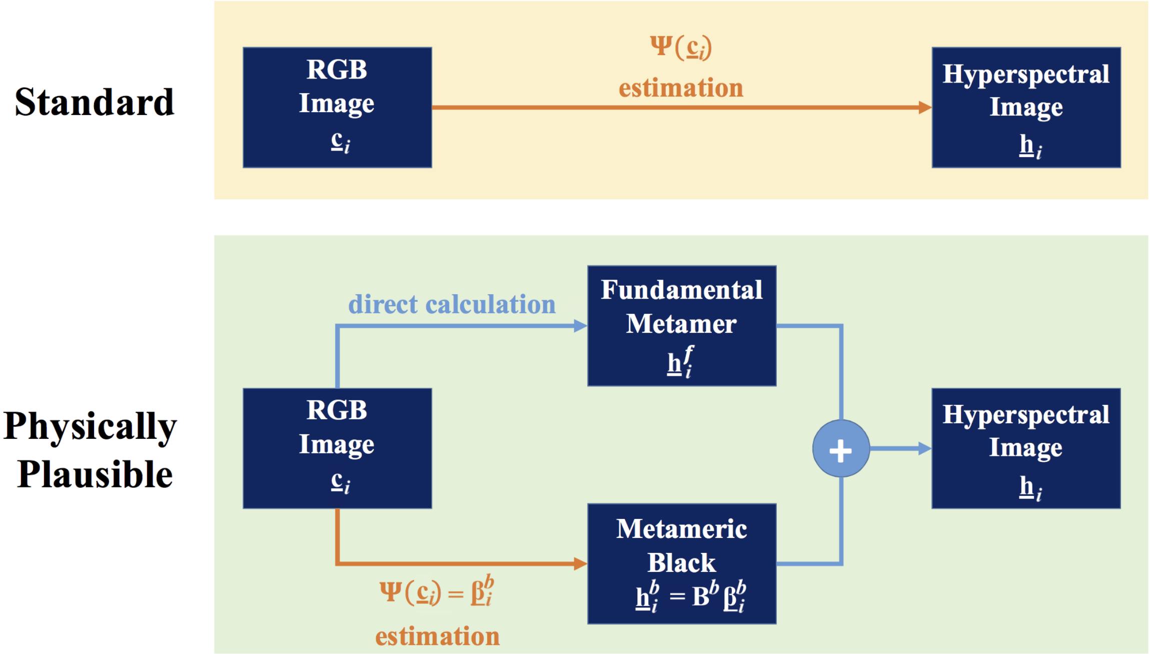

Standard
RGB Image $\underline{\mathbf{c}}_i$
$\Psi(\underline{\mathbf{c}}_i)$
estimation
Hyperspectral Image $\underline{\mathbf{h}}_i$
Physically Plausible
direct calculation
Fundamental Metamer $\underline{\mathbf{h}}_i^f$
RGB Image $\underline{\mathbf{c}}_i$
+
Hyperspectral Image $\underline{\mathbf{h}}_i$
Metameric Black $\underline{\mathbf{h}}_i^b = \mathbf{B}^b \underline{\boldsymbol{\beta}}_i^b$
$\Psi(\underline{\mathbf{c}}_i) = \underline{\boldsymbol{\beta}}_i^b$
estimation

Figure 7.3: The standard SR scheme (**top**) versus our physically plausible SR scheme (**bottom**).

the output layer is constricted to return positive values. Hence, we need to *offset* the ground-truth $\underline{\boldsymbol{\beta}}_i^b$ values in training so as to prevent the negative values for the DNNs to train properly.

Empirically, assume that the maximum value in the original hyperspectral images is v_{max} (e.g., in our case, ICVL hyperspectral images are 12-bit, so $v_{max} = 4095$), we found that the values in $\underline{\boldsymbol{\beta}}_i^b$ are typically bounded by $[-v_{max}, v_{max}]$. Without altering the original setup for the output layer of the DNNs, we set the algorithms to recover $\frac{1}{2}(\underline{\boldsymbol{\beta}}_i^b + v_{max})$, instead, and then correct the offset back from the predicted values in the inference phase. Here, the $\frac{1}{2}$ factor is to keep the values of the offset $\underline{\boldsymbol{\beta}}_i^b$ in the same range as the original ground-truth $\underline{\boldsymbol{h}}_i$.

Further, if we also wish to adopt a intensity-scaling data augmentation technique for the DNN (as detailed in Section 6.5), the range of augmentation will proportionally widen the range of $\underline{\boldsymbol{\beta}}_i^b$. For example, if we consider $\beta = 10$ in Equation (6.11) for data augmentation, the range of $\underline{\boldsymbol{\beta}}_i^b$ will become $[-10v_{max}, 10v_{max}]$ and so we will need to set the offset accordingly—making the algorithm to recover $\frac{1}{20}(\underline{\boldsymbol{\beta}}_i^b + 10v_{max})$ in training and correcting it back in inference.

7.3 Post-Facto Physical Plausibility Correction

Perforce, with the approach we proposed in Section 7.2, we still need to retrain the SR algorithms to ensure their physical plausibility. In this section, we explore the possibility to *post-process* the outputs from any pre-trained SR algorithms to make them physically plausible. Advantageously, this means that even for a "black-box" SR algorithm (where source code is not available), we are able to adopt this approach to ensure its physical plausibility.

7.3.1 Formulation

Given $\Psi(\underline{\boldsymbol{c}}_i)$, an SR-recovered spectrum from the input RGB $\underline{\boldsymbol{c}}_i$, again, let us separate it into:

$$\Psi(\underline{\boldsymbol{c}}_i) = \widehat{\underline{\boldsymbol{h}}}_i^f + \widehat{\underline{\boldsymbol{h}}}_i^b , \tag{7.16}$$

where $\widehat{\underline{\boldsymbol{h}}}_i^f$ and $\widehat{\underline{\boldsymbol{h}}}_i^b$ are respectively the fundamental metamer and the metameric black components of $\Psi(\underline{\boldsymbol{c}}_i)$.

We know that for an SR algorithm to be physically plausible, $\widehat{\underline{\boldsymbol{h}}}_i^f$ has to coincide with $\underline{\boldsymbol{h}}_i^f$ which is the fundamental metamer of the ground-truth, and $\underline{\boldsymbol{h}}_i^f$ can be calculated directly given the input $\underline{\boldsymbol{c}}_i$ and the RGB camera's spectral sensitivities $\mathbf{R}$ via Equation (7.9). Therefore, as a post-processing step, we can feasibly *replace* $\widehat{\underline{\boldsymbol{h}}}^f$, i.e., the original fundamental metamer of $\Psi(\underline{\boldsymbol{c}}_i)$, by $\underline{\boldsymbol{h}}^f$, which is the correct one calculated from $\underline{\boldsymbol{c}}_i$ via Equation (7.9). This process is regardless of whether the original $\widehat{\underline{\boldsymbol{h}}}^f$ is already correct (the same as $\underline{\boldsymbol{h}}^f$) or not. Mathematically, the process can be formulated as such:

$$\begin{aligned}\Psi'(\underline{\boldsymbol{c}}_i) &= \Psi(\underline{\boldsymbol{c}}_i) - \widehat{\underline{\boldsymbol{h}}}_i^f + \underline{\boldsymbol{h}}_i^f \\ &= \underline{\boldsymbol{h}}_i^f + \widehat{\underline{\boldsymbol{h}}}_i^b \ ; \quad \forall i \ .\end{aligned} \tag{7.17}$$

That is, we need simply to subtract the fundamental metamer from the SR-produced recovery, and then add the correct one back.

7.3.2 A Universal Improvement of RMSE Accuracy

Here, importantly, we are going to mathematically prove that, in Equation (7.17), $\Psi'(\underline{\boldsymbol{c}}_i)$ will always recover spectra with the same or better accuracy as those delivered by $\Psi(\underline{\boldsymbol{c}}_i)$.

Theorem: *Correcting the fundamental metamer of a spectral recovery will only improve or retain its RMSE accuracy.*

Proof: Let us write $\Delta_1 = ||\Psi(\underline{\boldsymbol{c}}_i) - \underline{\boldsymbol{h}}_i||_2^2$, which is the squared RMSE error of the original SR (ignoring $1/n$, the channel-averaging coefficient), and $\Delta_2 = ||\Psi'(\underline{\boldsymbol{c}}_i) - \underline{\boldsymbol{h}}_i||_2^2$, the squared RMSE of our new SR setup. We wish to prove that

$$\Delta_2 \leq \Delta_1 \ . \tag{7.18}$$

Using the nomenclatures of the respective fundamental metamer and metameric black splits for $\Psi(\underline{\boldsymbol{c}}_i)$ and $\Psi'(\underline{\boldsymbol{c}}_i)$ demonstrated in Equation (7.16) and (7.17), we can further derive Δ_2 and Δ_1. We have:

$$\Delta_2 = ||(\underline{\boldsymbol{h}}_i^f + \widehat{\underline{\boldsymbol{h}}}_i^b) - (\underline{\boldsymbol{h}}_i^f + \underline{\boldsymbol{h}}_i^b)||_2^2 = ||\widehat{\underline{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b||_2^2 \ . \tag{7.19}$$

and

$$\begin{aligned}\Delta_1 &= ||(\underline{\widehat{\boldsymbol{h}}}_i^f + \underline{\widehat{\boldsymbol{h}}}_i^b) - (\underline{\boldsymbol{h}}_i^f + \underline{\boldsymbol{h}}_i^b)||_2^2 \\ &= ||(\underline{\widehat{\boldsymbol{h}}}_i^f - \underline{\boldsymbol{h}}_i^f) + (\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b)||_2^2 \\ &= ||\underline{\widehat{\boldsymbol{h}}}_i^f - \underline{\boldsymbol{h}}_i^f||_2^2 + ||\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b||_2^2 + 2 \cdot [\underline{\widehat{\boldsymbol{h}}}_i^f - \underline{\boldsymbol{h}}_i^f]^\mathsf{T} [\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b] \ .\end{aligned} \tag{7.20}$$

Clearly,

$$\begin{cases} [\underline{\widehat{\boldsymbol{h}}}_i^f - \underline{\boldsymbol{h}}_i^f] \in \mathcal{C}(\mathbf{R}) \\ [\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b] \in \mathcal{C}(\mathbf{B}^b) \end{cases} , \tag{7.21}$$

where $\mathcal{C}()$ indicates the column space of a matrix. As $\mathcal{C}(\mathbf{R})$ and $\mathcal{C}(\mathbf{B}^b)$ are orthogonal spaces (Equation (7.8)), we get:

$$[\underline{\widehat{\boldsymbol{h}}}_i^f - \underline{\boldsymbol{h}}_i^f]^\mathsf{T} [\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b] = 0 \ . \tag{7.22}$$

Substituting into Equation (7.20):

$$\Delta_1 = ||\underline{\widehat{\boldsymbol{h}}}_i^f - \underline{\boldsymbol{h}}_i^f||_2^2 + ||\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b||_2^2 \ . \tag{7.23}$$

Clearly, jointly considering Equation (7.19) and (7.23), it is immediate that:

$$\Delta_2 = ||\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b||_2^2 \ \leq \ ||\underline{\widehat{\boldsymbol{h}}}_i^f - \underline{\boldsymbol{h}}_i^f||_2^2 + ||\underline{\widehat{\boldsymbol{h}}}_i^b - \underline{\boldsymbol{h}}_i^b||_2^2 = \Delta_1 \ ; \quad \forall i \ . \tag{7.24}$$

(*End of proof*).

Equation (7.24) encapsulates succinctly that the spectrum recovered via our proposed post-processing step is always as close or closer to the ground truth (compared to the original spectrum returned by any SR algorithm). Of course, we note that as our proof is tailored to RMSE, we can expect that this “universal” proof do not necessarily work for other metrics of concern, e.g., MRAE.

7.4 Experimental Results

In this section, we are going to show the results of several experiments comparing the algorithms under the **Original** training (where the algorithms are trained as the normal RGB-to-spectrum mappings), Physically Plausible Training (**P.P.T.**; Section 7.2), and Post-Facto Correction (**P.F.C.**; Section 7.3).

We consider the LS-based (see Section 4.2) regression methods, DNN-based HSCNN-R, and our new state-of-the-art sparse coding method A++. Again, for all experiments, the 4-fold cross-validation protocol as detailed in Section 3.1.3 is adopted, and the CIE 1964 colour matching functions [24] are used as the camera spectral sensitivity functions.

7.4.1 Colour Fidelity (Physical Plausibility) Test

First, we conduct a physical plausibility test as illustrated in Figure 7.4. While the ground-truth RGBs can be generated from the hyperspectral data (red curve follows red arrow), we test the colour fidelity of the SR-recovered spectra (blue dotted curve in the left panel) when reintegrated with the same set of spectral sensitivities (following the top blue arrow).

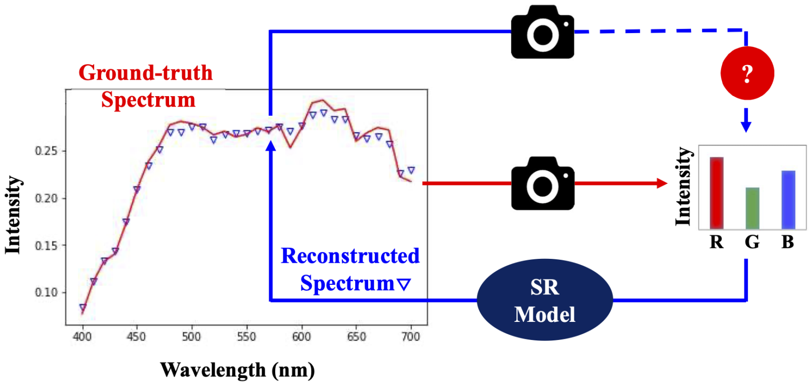


Figure 7.4: Our physical plausibility (colour fidelity) test for SR.

We use the CIE 2000 colour difference formula (ΔE_{00}) [78] to measure the difference between the ground-truth and reconstructed colours at individual pixels. The implementation of ΔE_{00} is rather complex: we refer the readers to [78] for details. Practically, a ΔE_{00} equalling

Table 7.1: Mean per-image mean and 99.9-percentile CIE ΔE 2000 colour errors introudced by the hyperspectral image reconstruction algorithms trained under the original, Physically Plausible Training (P.P.T.) and Post-Facto Correction (P.F.C.) setups.

	ΔE_{00}					
	Original		**P.P.T.**		**P.F.C.**	
	Mean	99.9 pt.	Mean	99.9 pt.	Mean	99.9 pt.
LR (LS)	0.05	0.79	0.00	0.00	0.00	0.00
RPR (LS)	0.14	1.48	0.00	0.00	0.00	0.00
A+ (LS)	0.06	2.47	0.00	0.00	0.00	0.00
RBFN (LS)	0.32	9.24	0.00	0.00	0.00	0.00
PR (LS)	0.01	0.18	0.00	0.00	0.00	0.00
HSCNN-R	0.10	2.06	0.00	0.00	0.00	0.00
A++	0.42	11.57	0.00	0.00	0.00	0.00

1 between two colour stimuli correlates with a colour difference that is just noticeable to a human observer.

Note that the ΔE_{00} is defined upon the CIELAB [71] colour coordinates—one of the standard (device independent) colour spaces [84]. Since the CIE 1964 colour matching functions were used to create the ground-truth RGBs, these RGBs are in effect the CIEXYZ tristimulus values. Then, from the CIEXYZ colours, there exists direct transformation to CIELAB given a ground-truth white point colour (i.e., the illumination colour) [84]. In our experiments, we obtained this white point information by hand-crafting the "brightest near-achromatic spectrum" from each ground-truth hyperspectral image and then integrating this white-surface radiance spectrum with the CIE 1964 XYZ colour matching functions.

In Table 7.1, we present the mean per-image mean and 99.9-percentile ΔE_{00} results. It is clear that all of the considered algorithms under their respective original training setups (under the "**Original**" headline in Table 7.1) introduce colour errors, with the extent varies depending on the algorithms. While the mean per-image-mean colour differences are shown to be all less than 1 (the just noticeable difference), we see that many of these algorithms suggest ΔE_{00}'s much larger than 1 at worst-case pixels (i.e., the per-image 99.9 percentile). However, again, we know that the very existence of colour errors indicates that all of these algorithms are not physically plausible, regardless of the magnitudes of the colour errors.

Then, we show that both our physically plausible training (**P.P.T.**) and post-facto correction (**P.F.C.**) approaches are able to ensure all algorithms to recover spectra of the exact same colours as the ground-truths—thus the 0 colour error under all circumstances. In other words, using either **P.P.T.** or **P.F.C.**, we are promised that the algorithms will not change the original input colours as they are used to recover spectra.

7.4.2 Spectral Accuracy

Of course, with the colour fidelity ensured, we also would like to see how incorporating P.P.T. and P.F.C. influences the spectral accuracy delivered by the algorithms. Like in previous chapters, we will present the spectral accuracy results in terms of mean per-image mean and 99-percentile MRAE errors, in Table 7.2. Additionally, the RMSE statistics are given in Table 7.3. We present the RMSE results because the P.F.C. approach, as proved in Section 7.3.2, is able to improve the RMSE error (but not MRAE) for each and every reconstructed spectrum. Remember that while MRAE can be understood as a percentage error, RMSE is calculated at an absolute scale where, here, the maximum pixel value at each spectral channel is 4095 (for 12-bit hyperspectral images).

First, let us look at the MRAE results (Table 7.2). We see that adopting P.P.T. or P.F.C. does not change much of the original performance of the tested algorithms. This means we can, in effect, recover spectra of perfect colour fidelity using P.P.T. and P.F.C. without deteriorating the spectral accuracy. With P.F.C., it is shown that we further advance the state-of-the-art MRAE performance. Indeed, the A++ algorithm with P.F.C. delivers slightly better per-image-mean results compared to the original A++, and HSCNN-R, which delivers the best per-image-99-percentile results, is also slightly but generally improved using P.F.C..

Table 7.2: Mean per-image mean and 99-percentile MRAE errors introudced by the hyperspectral image reconstruction algorithms trained under the original, Physically Plausible Training (P.P.T.) and Post-Facto Correction (P.F.C.) setups.

Method	**MRAE (%)**					
	Original		**P.P.T.**		**P.F.C.**	
	Mean	99 pt.	Mean	99 pt.	Mean	99 pt.
LR (LS)	6.24	16.95	6.23	17.07	6.25	16.98
RPR (LS)	4.69	16.97	4.60	15.57	4.64	15.40
A+ (LS)	3.81	15.52	3.83	14.93	3.86	15.25
RBFN (LS)	2.06	7.89	1.96	7.48	1.98	7.47
PR (LS)	1.95	7.10	1.94	7.09	1.95	7.11
HSCNN-R	1.73	6.53	1.76	6.65	1.72	6.49
A++	1.69	7.78	1.71	7.77	1.68	7.62

Next, the RMSE results (Table 7.3). Let us firstly compare the performance of the algorithms under original training and P.F.C. setup. We see the mean per-image-mean RMSE performances are always better when P.F.C. is adopted, for all the algorithms considered. This is in accordance with the proof we provided in Section 7.3.2.

As for the 99-percentile performance, in most cases P.F.C. does also improve the performance, however we see for the PR (LS) algorithm, the performance is very slightly degraded. We note that this outcome is not in contradiction with our proof in Section 7.3.2, since different pixels can correspond to the 99 percentile of an image before and after P.F.C. is adopted, while the proof only ensures the improvement at the exact same pixel.

Then, let us now also consider the RMSE performance delivered by P.P.T.. Evidently, unlike P.F.C., P.P.T. does not ensure the definite improvement of RMSE (as we can see the mean per-image-mean RMSE does not always improve from the original training when using P.P.T.). Still, usefully, we see that P.P.T. can occasionally provide even better RMSE performance than P.F.C., e.g., the mean performance of LR, RPR, A+ and RBFN (also on average, considering all methods, P.P.T. performs slightly better than P.F.C.).

Table 7.3: Mean per-image mean and 99-percentile RMSE errors introudced by the hyperspectral image reconstruction algorithms trained under the original, Physically Plausible Training (P.P.T.) and Post-Facto Correction (P.F.C.) setups.

Method	**RMSE**					
	Original		**P.P.T.**		**P.F.C.**	
	Mean	99 pt.	Mean	99 pt.	Mean	99 pt.
LR (LS)	33.26	99.76	33.23	99.75	33.25	99.72
RPR (LS)	27.80	94.64	27.49	95.96	27.54	94.13
A+ (LS)	23.97	94.18	24.36	97.03	23.86	93.77
RBFN (LS)	18.30	81.88	17.50	77.41	17.56	77.47
PR (LS)	17.05	75.56	17.06	75.59	17.04	75.57
HSCNN-R	16.33	77.21	16.34	77.38	16.26	77.07
A++	15.43	86.18	15.54	86.89	15.31	84.04

7.5 Summary

Regression- and DNN-based algorithms usually only concern about minimising spectral accuracy error, but the underlying physical relationship between spectra and colours is not preserved. This type of physically *implausible* mapping causes the issue of poor colour fidelity (i.e., the input RGB colours are different from the ones corresponding to the recovered spectra). For some algorithms, e.g., RBFN and A++, the colour shift can be very significant at the per-image 99.9-percentile (worst-case) pixels.

In this chapter we show that, with respect to a given input RGB, all spectral *candidates* for a physically plausible SR can be represented by a *fixed* fundamental metamer defined by a linear combination of camera spectral sensitivities, and a metameric black which does not contribute to the colour formation.

Relative to this insight, our first physically plausible spectral recovery approach sets out to reconstruct only the metameric black's basis coefficients from the RGBs, while the fundamental metamers are *derived* (from the input RGBs) directly. In our second approach, we do not retrain the algorithms but only replace the fundamental metamers of the recovered spectra by the derived (correct) ones. While both methods are effective for ensuring perfect colour fidelity of the tested SR algorithms, the latter method—the Post-Facto Correction approach—is proved to improve the RMSE spectral accuracy of *every* recovered spectrum.

Chapter 8

Conclusion

In this thesis, we studied the relative performance of regression- and DNN-based spectral reconstruction (SR) algorithms. While the former is much simpler than the latter, we found that the best performing regression—before any advances made in this thesis—was merely 12% worse than the leading DNN method.

With a pioneer study on using different cameras for SR, we learned that some cameras are able to support much better SR compared to other cameras. Significantly, we showed that a better performing camera for a much simpler algorithm (e.g., polynomial regression) can sometimes surpass the performance of the more complex algorithm (e.g., the DNN) implemented on a worse camera model.

We also proposed a real-world worst-case imaging condition for SR called Radiance Mondrian World (RMW). Under RMW, the images are constructed such that there is no meaningful image content and also no one radiance spectrum has higher chance to appear than another. We tested SR algorithms of different levels of complexity and discovered that they all degrade (from their original performance) to around the same level of performance when tested on the RMW images. Moreover, even when we retrained the algorithms on an RMW training set, we saw no benefit of using a complex DNN method rather than the simple linear regression algorithm.

Further, we attempted to advance the regression-based SR methods. We realised that regression methods used in SR almost always solve for the closed-form least-squares minimisation, where the squared Root-Mean-Square Error (RMSE) is minimised, whereas the top DNN methods are more commonly evaluated and ranked using the Mean Relative Absolute Error (MRAE). We also found that conventional regressions regularise the mappings in all spectral channels together, but in regression the values in each spectral channel are estimated from the RGBs independently of the other channels. By reformulating regressions such that they minimise metrics more similar to MRAE and regularise per spectral channel, we made regressions more competitive against the leading DNNs under the MRAE evaluation, now with only a 9% performance gap between the reformulated polynomial regression and the leading DNN.

Our second attempt of advancing regressions incorporated the *sparse coding* strategy. While most of the sparse coding methods localise SR mappings in the RGB neighbourhoods, we found that doing so in the spectral space can deliver much better performance. First, given an input query RGB, we used our reformulated polynomial regression SR to estimate the spectral neighbourhood its corresponding ground-truth spectrum might locate in. Then, for each neighbourhood, a linear regression SR map was used. This setup not only advances the original polynomial regression's performance, but even reaches the state-of-the-art performance—surpassing the leading DNN method.

This thesis also contributes to identifying practical issues of training-based SR algorithms in general. First, the exposure invariance problem. We showed that leading SR algorithms are usually exposure non-invariant, i.e., algorithms that can only work under fixed exposure condition while degrades as the testing scenes get dimmer or brighter. Given this finding, we proposed an exposure invariant "root-polynomial regression" SR method which has superior performance at the worst-case pixels of the images among other exposure invariant alternatives.

We further proposed two approaches that can effectively enforce exposure invariance upon existing SR algorithms. The *chromaticity mapping* approach separates the input colours into two multiplying terms: chromaticity and brightness, and operates SR algorithm only on the chromaticity term. Then, in the *data augmentation* approach, we randomly perturb the brightness of the data the algorithms are trained on. We showed that regression-based methods work better with chromaticity mapping, whereas the leading DNN incorporated with data augmentation provides the best exposure invariant performance overall.

Our last contribution is studying the *physical plausibility* of SR algorithms, i.e., whether the algorithms-recovered spectra preserve the known physical relation between ground-truth spectra and input RGBs. As we found neither regressions nor leading DNNs are physically plausible, we proposed two general approaches to enforce this property. Both approaches are based on the idea that each spectrum can be separated into two additive terms, the *fundamental metamer* and the *metameric black*. The former term is a linear combination of the RGB camera's three spectral sensitivity functions and is fixed given the input RGB, whereas the latter term is not bounded by the input RGB.

In the first approach, the Physically Plausible Training (P.P.T.), we retrained the SR algorithms to recover only the metameric blacks from the input RGBs, and later adding the correctly calculated fundamental metamers to the estimated metameric blacks to derive the final spectral recovery. Then, in the second approach we do not retrain the algorithms. Instead, we follow a Post-Facto Correction (P.F.C.) process by *replacing* the potentially-wrong fundamental metameras of the recovered spectra with the calculated ones.

Apart from their effectiveness in preserving the physical plausibility of the SR algorithms, it was proved mathematically that the P.F.C. approach will always improve the RMSE spectral accuracy of the original algorithms. While we do not have the same proof for P.P.T., it was shown that P.P.T. can occasionally deliver better SR accuracy than P.F.C..

8.1 Future Work

Several interesting research ideas can be extended from this thesis:

- In our test, the best camera model for different SR algorithms can be different. Then, if we allow the camera sensitivity functions to be free variables, we might be able to *customise* a camera model for a particular SR algorithm. This is eminently achievable for DNNs where both camera sensitivity functions and SR can be optimised together in a single framework.

- The Radiance Mondrian World (RMW) defines a worst-case imaging condition for SR, under which the DNNs perform on par with linear regression (i.e., we do not get better RMW performance by increasing the model complexity). It will be interesting to investigate if any SR algorithm can improve the SR performance on RMW images.
- In the first step of A++, we predict the ground-truth spectrum's whereabouts in the spectral neighbourhoods using another SR algorithm, PR-RELS. As A++ now performs better than PR-RELS, we could in turn use A++ to predict the ground-truth spectral neighbourhoods instead of PR-RELS (and, if the resulting method is further improved, we could then use this new method instead of A++ to predict the spectral neighbourhoods). However, we could expect the training and inference complexity will significantly increase at every new iteration.
- In the prior art, almost all DNN-based SR methods use image patches as input. With the development of A++, we show that a pixel-based mapping is all we need to reach the current state-of-the-art SR performance. Still, DNN is a powerful learning algorithm. Even without mapping image patches, we should be able to construct a "pixel-based DNN" which could potentially outperform A++.
- DNN-like (patch-based) SR mapping still has merit: it is possible to distinguish different spectra of the same input colour using the colours of the surrounding pixels. In some applications where this ability is necessary, given a regression algorithm, we could also consider to *regress a small image patch.* E.g., if a 3×3 patch is considered, we could stack all 9 RGBs into a longer input vector for linear regression. To study the effectiveness of this setup, we will also need a more targeted image database where same-coloured spectra are of greater abundance and importance (e.g., real and fake objects, real human face vs. a poster, etc.).

- Both chromaticity mapping and data augmentation—our attempts of enforcing exposure invariance to the originally exposure non-invariant SR algorithms—make the algorithms perform worse than how they perform under fixed exposure. It could be because in chromaticity mapping we remove the brightness variation from the input data (which could potentially be useful to distinguish spectra), and in data augmentation the algorithms have larger data variation to learn. It is possible that we now need to increase the model complexity, e.g., higher polynomial order for polynomial regression or deeper network architecture for the DNNs, to tackle the more difficult learning problems they have become.
- Our Post-Facto Correction (P.F.C.) for physically plausible SR is proved to universally improve spectral accuracy in terms of RMSE. However, we have argued that RMSE is less suitable for evaluating SR compared to the relative errors such as MRAE. A next challenge could be to develop a P.F.C. approach which provably improve the MRAE accuracy.
- In this thesis we looked at spectral reconstruction from the 3-sensor RGB images. We could evaluate SR methods developed with more than 3 sensors, and test the hypothesis that the marginal benefit of patch-based processing will diminish as the number of sensors increase.

Glossary

chromaticitiy colour divided by brightness.

CIE 2000 colour difference a modern colour difference formula aiming to match its unit with the just noticeable colour difference of a standard human observer.

CIELAB a standard colour coordinate system where the Euclidean distance calculated between coordinates is commonly accepted as a perceptually uniform colour difference measure (but much less uniform than CIE 2000 colour difference formula).

CIEXYZ a standard colour coordinate system where CIE colour matching functions are the underlying spectral sensitivities.

colour matching functions a set of three spectral functions that are a linear transform away from the spectral sensitivities of the three types of human cone cells.

data augmentation a technique to improve a learning model's generalisability by perturbing the training data.

exposure invariance the property of a spectral reconstruction algorithm where an input scaled by a scalar exposure factor returns an output scaled by the same factor.

fundamental metamer the component of a spectrum that lies in the spectral subspace spanned by the camera spectral sensitivities.

hyperspectral camera a camera that records high-resolution spectra at every pixel of an image.

illuminant light source.

metamer a member of the same-coloured spectra under a given viewing condition.

metameric black the component of a spectrum perpendicular to the spectral subspace spanned by the camera spectral sensitivities.

metamerism the phenomenon where different spectra having the same colour under one viewing condition can become different in colour under other viewing conditions.

multispectral camera a camera that records multiple spectral channels at each pixel of an image.

oracle solution a theoretical method that assumes perfect decisions in the process.

physical plausibility the property of a spectral reconstruction algorithm where the recovered spectra can reproduce the input RGBs via the underlying physical process.

radiance the combined spectral signal of light source spectrum and surface reflectance.

reflectance a wavelength-dependent ratio function referring to the reflected light intensity divided by the incident light intensity at each wavelength; it is an intrinsic property of an object surface.

regularisation adjusting the bias of the fitting function to overcome the overfitting problem of a learning model.

relative error the deviation measured as a percentage with respect to the ground-truth values.

spectral reconstruction a process that recovers hyperspectral measurements from RGB or multispectral measurements.

spectral sensitivity the wavelength-dependent sensor response function.

spectral signature the concept that light signals coming from different sources and/or reflected from different materials can be distinguished by their spectral features.

viewing condition the lighting and/or camera sensor under which a spectral signal is viewed.

Bibliography

[1] Aeschbacher, J., Wu, J., and Timofte, R. In defense of shallow learned spectral reconstruction from RGB images. In *Proceedings of the International Conference on Computer Vision* (2017), IEEE, pp. 471–479.

[2] Agahian, F., Amirshahi, S. A., and Amirshahi, S. H. Reconstruction of reflectance spectra using weighted principal component analysis. *Color Research & Application 33*, 5 (2008), 360–371.

[3] Aharon, M., Elad, M., and Bruckstein, A. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing 54*, 11 (2006), 4311–4322.

[4] Alvarez-Gila, A., Van De Weijer, J., and Garrote, E. Adversarial networks for spatial context-aware spectral image reconstruction from RGB. In *Proceedings of the International Conference on Computer Vision* (2017), IEEE, pp. 480–490.

[5] Arad, B., and Ben-Shahar, O. Sparse recovery of hyperspectral signal from natural RGB images. In *Proceedings of the European Conference on Computer Vision* (2016), Springer, pp. 19–34.

[6] Arad, B., and Ben-Shahar, O. Filter selection for hyperspectral estimation. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 3153–3161.

[7] 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* (2018), IEEE, pp. 929–938.

[8] Arad, B., Timofte, R., Ben-Shahar, O., Lin, Y., Finlayson, G., et al. NTIRE 2020 challenge on spectral reconstruction from an RGB image. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops* (2020), IEEE.

[9] Arad, B., Timofte, R., Yahel, R., Morag, N., Bernat, A., Cai, Y., Lin, J., Lin, Z., Wang, H., Zhang, Y., et al. Ntire 2022 spectral recovery challenge and data set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 863–881.

[10] Arguello, H., and Arce, G. R. Colored coded aperture design by concentration of measure in compressive spectral imaging. *IEEE Transactions on Image Processing 23*, 4 (2014), 1896–1908.

[11] Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software 22*, 4 (1996), 469–483.

[12] Benoudjit, N., Archambeau, C., Lendasse, A., Lee, J. A., Verleysen, M., et al. Width optimization of the gaussian kernels in radial basis function networks. In *Proceedings of the European Symposium on Artificial Neural Networks* (2002), vol. 2, pp. 425–432.

[13] Bianco, S. Reflectance spectra recovery from tristimulus values by adaptive estimation with metameric shape correction. *Journal of the Optical Society of America A 27*, 8 (2010), 1868–1877.

[14] Brainard, D., and Freeman, W. Bayesian color constancy. *Journal of the Optical Society of America A 14*, 7 (1997), 1393–1411.

[15] Brauers, J., Schulte, N., and Aach, T. Multispectral filter-wheel cameras: Geometric distortion model and compensation algorithms. *IEEE Transactions on Image Processing 17*, 12 (2008), 2368–2380.

[16] Buchsbaum, G., and Gottschalk, A. Trichromacy, opponent colours coding and optimum colour information transmission in the retina. In *Proceedings of the Royal society of London. Series B. Biological Sciences* (1983), vol. 220, The Royal Society London, pp. 89–113.

[17] Cao, X., Du, H., Tong, X., Dai, Q., and Lin, S. A prism-mask system for multispectral video acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 12 (2011), 2423–2435.

[18] Carroll, R. J., and Ruppert, D. *Transformation and Weighting in Regression*, vol. 30. CRC Press, 1988.

[19] Chen, K., Lv, Q., Lu, Y., and Dou, Y. Robust regularized extreme learning machine for regression using iteratively reweighted least squares. *Neurocomputing 230* (2017), 345–358.

[20] Chen, S., Cowan, C. F., and Grant, P. M. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks 2*, 2 (1991), 302–309.

[21] Chen, Z., Wang, J., Wang, T., Song, Z., Li, Y., Huang, Y., Wang, L., and Jin, J. Automated in-field leaf-level hyperspectral imaging of corn plants using a cartesian robotic platform. *Computers and Electronics in Agriculture 183* (2021), 105996.

[22] Cheney, W., and Kincaid, D. *Linear Algebra: Theory and Applications*, vol. 110. The Australian Mathematical Society, 2009.

[23] Cheung, V., Westland, S., Li, C., Hardeberg, J., and Connah, D. Characterization of trichromatic color cameras by using a new multispectral imaging technique. *Journal of the Optical Society of America A 22*, 7 (2005), 1231–1240.

[24] Commission Internationale de L'eclairage. *CIE Proceedings (1964) Vienna Session, Committee Report E-1.4. 1* (1964).

[25] Connah, D., and Hardeberg, J. Spectral recovery using polynomial models. In *Proceedings of SPIE Color Imaging X: Processing, Hardcopy, and Applications* (2005), vol. 5667, International Society for Optics and Photonics, pp. 65–75.

[26] Courtenay, L., 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., et al. Hyperspectral imaging and robust statistics in non-melanoma skin cancer analysis. *Biomedical Optics Express 12*, 8 (2021), 5107–5127.

[27] Dong, W., Zhou, C., Wu, F., Wu, J., Shi, G., and Li, X. Model-guided deep hyperspectral image super-resolution. *IEEE Transactions on Image Processing 30* (2021), 5754–5768.

[28] Finlayson, G., Mackiewicz, M., and Hurlbert, A. Color correction using root-polynomial regression. *IEEE Transactions on Image Processing 24*, 5 (2015), 1460–1470.

[29] Finlayson, G., and Morovic, P. Metamer sets. *Journal of the Optical Society of America A 22*, 5 (2005), 810–819.

[30] Fu, Y., Zhang, T., Zheng, Y., Zhang, D., and Huang, H. Joint camera spectral response selection and hyperspectral image recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 1 (2020), 256–272.

[31] Galatsanos, N., and Katsaggelos, A. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Transactions on Image Processing 1*, 3 (1992), 322–336.

[32] Galliani, S., Lanaras, C., Marmanis, D., Baltsavias, E., and Schindler, K. Learned spectral super-resolution. *arXiv preprint arXiv:1703.09470* (2017).

[33] Galvis, L., Lau, D., Ma, X., Arguello, H., and Arce, G. R. Coded aperture design in compressive spectral imaging based on side information. *Applied Optics 56*, 22 (2017), 6332–6340.

[34] Garcia, H., Correa, C. V., and Arguello, H. Multi-resolution compressive spectral imaging reconstruction from single pixel measurements. *IEEE Transactions on Image Processing 27*, 12 (2018), 6174–6184.

[35] Gentle, J. E. Matrix algebra. *Springer Texts in Statistics 10* (2007), 232–233.

[36] Gomes, V., Mendes-Ferreira, A., and Melo-Pinto, P. Application of hyperspectral imaging and deep learning for robust prediction of sugar and ph levels in wine grape berries. *Sensors 21*, 10 (2021), 3459.

[37] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning.* MIT press, 2016.

[38] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM 63*, 11 (2020), 139–144.

[39] Grillini, F., Thomas, J., and George, S. Mixing models in close-range spectral imaging for pigment mapping in cultural heritage. In *Proceedings of the International Colour Association (AIC) Conference* (2020), pp. 372–376.

[40] Hao, X., Funt, B., and Jiang, H. Evaluating colour constancy on the new mist dataset of multi-illuminant scenes. In *Proceedings of the Color and Imaging Conference* (2019), vol. 2019, Society for Imaging Science and Technology, pp. 108–113.

[41] Hardeberg, J. Y. On the spectral dimensionality of object colours. In *Proceedings of the Conference on Colour in Graphics, Imaging, and Vision* (2002), vol. 2002, Society for Imaging Science and Technology, pp. 480–485.

[42] Heikkinen, V., Lenz, R., Jetsu, T., Parkkinen, J., Hauta-Kasari, M., and 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 (2008), 2444–2458.

[43] Hu, J., Jia, X., Li, Y., He, G., and Zhao, M. Hyperspectral image superresolution via intrafusion network. *IEEE Transactions on Geoscience and Remote Sensing 58*, 10 (2020), 7459–7471.

[44] Hunt, G. R. Spectral signatures of particulate minerals in the visible and near infrared. *Geophysics 42*, 3 (1977), 501–513.

[45] Jiang, J., Liu, D., Gu, J., and Süsstrunk, S. What is the space of spectral sensitivity functions for digital color cameras? In *2013 IEEE Workshop on Applications of Computer Vision* (2013), IEEE, pp. 168–179.

[46] Joslyn Fubara, B., Sedky, M., and Dyke, D. RGB to spectral reconstruction via learned basis functions and weights. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops* (2020), IEEE, pp. 480–481.

[47] Kaya, B., Can, Y., and Timofte, R. Towards spectral estimation from a single RGB image in the wild. In *Proceedings of the International Conference on Computer Vision Workshop* (2019), IEEE, pp. 3546–3555.

[48] Kerekes, J. P., and Schott, J. R. Hyperspectral imaging systems. *Hyperspectral Data Exploitation: Theory and Applications* (2007), 19–45.

[49] Kokoska, S., and Zwillinger, D. *CRC Standard Probability and Statistics Tables and Formulae.* CRC Press, 2000.

[50] Land, E. The retinex theory of color vision. *Scientific american 237*, 6 (1977), 108–129.

[51] Lee, T.-W., Wachtler, T., and Sejnowski, T. J. The spectral independent components of natural scenes. In *Proceedings of the International Workshop on Biologically Motivated Computer Vision* (2000), Springer, pp. 527–534.

[52] Lever, J., Krzywinski, M., and Altman, N. Points of significance: model selection and overfitting. *Nature Methods 13*, 9 (2016), 703–705.

[53] Li, J., Wu, C., Song, R., Li, Y., and 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* (2020), IEEE, pp. 462–463.

[54] Li, X., Zhao, C., and Yang, Y. Hyperspectral anomaly detection based on the distinguishing features of a redundant difference-value network. *International Journal of Remote Sensing 42*, 14 (2021), 5459–5477.

[55] Lin, X., Liu, Y., Wu, J., and Dai, Q. Spatial-spectral encoded compressive hyperspectral imaging. *ACM Transactions on Graphics 33*, 6 (2014), 233.

[56] Lv, M., Chen, T., Yang, Y., Tu, T., Zhang, N., Li, W., and Li, W. Membranous nephropathy classification using microscopic hyperspectral imaging and tensor

patch-based discriminative linear regression. *Biomedical Optics Express 12*, 5 (2021), 2968–2978.

[57] MacQueen, J. Classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (1967), pp. 281–297.

[58] Maloney, L., and Wandell, B. Color constancy: a method for recovering surface spectral reflectance. *Journal of the Optical Society of America A 3*, 1 (1986), 29–33.

[59] Marimont, D. H., and Wandell, B. A. Linear models of surface and illuminant spectra. *Journal of the Optical Society of America A 9*, 11 (1992), 1905–1913.

[60] Morovic, P., and Finlayson, G. Metamer-set-based approach to estimating surface reflectance from camera RGB. *Journal of the Optical Society of America A 23*, 8 (2006), 1814–1822.

[61] Ng, A. Whether to use end-to-end deep learning. https://www.youtube.com/watch?v=l_-CUyEx_x4&ab_channel=DeepLearningAI, 2017. Accessed: 2023-05-31.

[62] Nguyen, R., Prasad, D., and Brown, M. Training-based spectral reconstruction from a single RGB image. In *Proceedings of the European Conference on Computer Vision* (2014), Springer, pp. 186–201.

[63] Oh, W. S., Brown, M. S., Pollefeys, M., and Joo Kim, S. Do it yourself hyperspectral imaging with everyday digital cameras. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (2016), IEEE, pp. 2461–2469.

[64] Parkkinen, J. P., Hallikainen, J., and Jaaskelainen, T. Characteristic spectra of munsell colors. *Journal of the Optical Society of America A 6*, 2 (1989), 318–322.

[65] Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers* (1993), IEEE, pp. 40–44.

[66] Penrose, R. A generalized inverse for matrices. In *Mathematical Proceedings of the Cambridge Philosophical Society* (1955), vol. 51, Cambridge University Press, pp. 406–413.

[67] Picollo, M., Cucci, C., Casini, A., and Stefani, L. Hyper-spectral imaging technique in the cultural heritage field: New possible scenarios. *Sensors 20*, 10 (2020), 2843.

[68] Ramanath, R., Snyder, W. E., Bilbro, G. L., and Sander, W. A. Demosaicking methods for bayer color arrays. *Journal of Electronic Imaging 11*, 3 (2002), 306–315.

[69] Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., and Baker, T. Analysis of dimensionality reduction techniques on big data. *IEEE Access 8* (2020), 54776–54788.

[70] Ribés, A., and Schmit, F. Reconstructing spectral reflectances with mixture density networks. In *Proceedings of the Conference on Colour in Graphics, Imaging, and Vision* (2002), vol. 2002, Society for Imaging Science and Technology, pp. 486–491.

[71] Robertson, A. The CIE 1976 color-difference formulae. *Color Research & Application 2*, 1 (1977), 7–11.

[72] Romero, J., Garcia-Beltrán, A., and Hernández-Andrés, J. Linear bases for representation of natural and artificial illuminants. *Journal of the Optical Society of America A 14*, 5 (1997), 1007–1014.

[73] Rousseeuw, P. J., and Croux, C. Alternatives to the median absolute deviation. *Journal of the American Statistical Association 88*, 424 (1993), 1273–1283.

[74] Rueda, H., Arguello, H., and Arce, G. R. Dmd-based implementation of patterned optical filter arrays for compressive spectral imaging. *Journal of the Optical Society of America A 32*, 1 (2015), 80–89.

[75] Schlossmacher, E. An iterative technique for absolute deviations curve fitting. *Journal of the American Statistical Association 68*, 344 (1973), 857–859.

[76] Schneider, T., Young, R., Bergen, T., Dam-Hansen, C., Goodman, T., Jordan, W., Lee, D.-H., Okura, T., Sperfeld, P., Thorseth, A., et al. CIE 250: 2022 spectroradiometric measurement of optical radiation sources. CIE-International Commission on Illumination, 2022.

[77] Sharma, G., and Wang, S. Spectrum recovery from colorimetric data for color reproductions. In *SPIE Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII* (2001), vol. 4663, International Society for Optics and Photonics, pp. 8–14.

[78] Sharma, G., Wu, W., and Dalal, E. N. The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. *Color Research & Application 30*, 1 (2005), 21–30.

[79] Sharma, S., Sharma, S., and Athaiya, A. Activation functions in neural networks. *Towards Data Science 6*, 12 (2017), 310–316.

[80] Shi, Z., Chen, C., Xiong, Z., Liu, D., and Wu, F. Hscnn+: Advanced cnn-based hyperspectral recovery from RGB images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops* (2018), IEEE, pp. 939–947.

[81] Snedecor, G., and Cochran, W. *Statistical Methods 6th Edition.* The Iowa State University, Ames, Iowa, 1967.

[82] Stiebel, T., and Merhof, D. Brightness invariant deep spectral super-resolution. *Sensors 20*, 20 (2020), 5789.

[83] Strang, G. *Introduction to Linear Algebra, 5th Edition.* Wellesley, MA: Wellesley-Cambridge Press, 2016.

[84] Süsstrunk, S., Buckley, R., and Swen, S. Standard RGB color spaces. In *Proceedings of Color and Imaging Conference* (1999), vol. 1999, Society for Imaging Science and Technology, pp. 127–134.

[85] Takatani, T., Aoto, T., and Mukaigawa, Y. One-shot hyperspectral imaging using faced reflectors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (2017), IEEE, pp. 4039–4047.

[86] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*, 1 (1996), 267–288.

[87] Tikhonov, A., Goncharsky, A., Stepanov, V., and Yagola, A. *Numerical Methods for the Solution of Ill-posed Problems*, vol. 328. Springer Science & Business Media, 2013.

[88] Timofte, R., De Smet, V., and Van Gool, L. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Proceedings of Asian Conference on Computer Vision* (2014), Springer, pp. 111–126.

[89] Tofallis, C. Least squares percentage regression. *Journal of Modern Applied Statistical Methods* (2009).

[90] Torun, O., and Yuksel, S. Unsupervised segmentation of lidar fused hyperspectral imagery using pointwise mutual information. *International Journal of Remote Sensing 42*, 17 (2021), 6461–6476.

[91] Van Trigt, C. Smoothest reflectance functions. ii. complete results. *Journal of Optical Society of America A 7*, 12 (1990), 2208–2222.

[92] Wagner, H. M. Linear programming techniques for regression analysis. *Journal of the American Statistical Association 54*, 285 (1959), 206–212.

[93] Wandell, B. The synthesis and analysis of color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1 (1987), 2–13.

[94] Wang, C., Wang, X., and Hardeberg, J. Y. A linear interpolation algorithm for spectral filter array demosaicking. In *Proceedings of International Conference on Image and Signal Processing* (2014), Springer, pp. 151–160.

[95] Wang, L., Gordon, M. D., and Zhu, J. Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In *Proceedings of the International Conference on Data Mining* (2006), IEEE, pp. 690–700.

[96] Wang, L., Xiong, Z., Gao, D., Shi, G., and Wu, F. Dual-camera design for coded aperture snapshot spectral imaging. *Applied Optics 54*, 4 (2015), 848–858.

[97] Wang, W., Ma, L., Chen, M., and Du, Q. Joint correlation alignment-based graph neural network for domain adaptation of multitemporal hyperspectral remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14* (2021), 3170–3184.

[98] Webb, G. I. Overfitting. In *Encyclopedia of Machine Learning* (2010), Springer US, pp. 744–744.

[99] Wyszecki, G., and Stiles, W. S. *Color Science*, vol. 8. Wiley New York, 1982.

[100] Xu, P., Xu, H., Diao, C., and Ye, Z. Self-training-based spectral image reconstruction for art paintings with multispectral imaging. *Applied Optics 56*, 30 (2017), 8461–8470.

[101] Yasuma, F., Mitsunaga, T., Iso, D., and Nayar, S. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing 19*, 9 (2010), 2241–2253.

[102] Zhang, X., Ma, X., Huyan, N., Gu, J., Tang, X., and Jiao, L. Spectral-difference low-rank representation learning for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing* (2021).

[103] Zhao, Y., and Berns, R. S. Image-based spectral reflectance reconstruction using the matrix r method. *Color Research & Application 32*, 5 (2007), 343–351.

[104] Zhao, Y., Guo, H., Ma, Z., Cao, X., Yue, T., and Hu, X. Hyperspectral imaging with random printed mask. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (2019), IEEE, pp. 10149–10157.

[105] Zhao, Y., Po, L.-M., Yan, Q., Liu, W., and Lin, T. Hierarchical regression network for spectral reconstruction from RGB images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops* (2020), IEEE, pp. 422–423.

[106] Zou, H., and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*, 2 (2005), 301–320.

[107] Zuffi, S., Santini, S., and Schettini, R. From color sensor space to feasible reflectance spectra. *IEEE Transactions on Signal Processing 56*, 2 (2008), 518–531.

[108] Zuzak, K. J., Francis, R. P., Wehner, E. F., Smith, J., Litorja, M., Allen, D. W., Tracy, C., Cadeddu, J., and Livingston, E. Hyperspectral imaging utilizing lctf and dlp technology for surgical and clinical applications. In *Proceedings of SPIE Design and Quality for Biomedical Technologies II* (2009), vol. 7170, pp. 71–79.