https://doi.org/10.2352/CIC.2022.30.1.37

Evaluating the Performance of Different Cameras for Spectral Reconstruction

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

Abstract

Spectral Reconstruction (SR) algorithms seek to map RGB images to their hyperspectral image counterparts. Statistical methods such as regression, sparse coding and deep neural networks are used to determine the SR mapping. All these algorithms are optimized 'blindly' and the provenance of the RGBs is not considered.

In this paper, we benchmark the performance of SR methods—in order of increasing complexity: regression, sparse coding, and deep neural network—when different RGB camera spectral sensitivity functions are used. In effect, we ask: "Are some cameras better able to recover spectra from RGBs than others?". In our experiments, RGB images are generated by numerical integration for a fixed set of hyperspectral images using 9 different camera response functions (each from a different camera manufacturer) plus the CIE 1964 color matching functions. Then, we train SR methods on the respective RGB image sets. Our experiments show three important results. First, different cameras **do** *support slightly better or worse spectral reconstruction but, secondly, that changing the spectral sensitivities alone does not change the ranking of different algorithms. Finally, we show that sometimes switching the used camera for SR can give a greater performance boost than switching to use a more complex SR method.*

1. Introduction

The light radiance coming from the scene has varying intensities at each wavelength, namely the light 'spectrum'. This information provides rich descriptions of the light source and object surfaces (from which the illuminating light reflects), and it is considered useful for various computer vision and computer graphics applications. Examples include remote sensing [40, 38], medical imaging [28, 13], device color calibration [11, 33] and art conservation [31, 19]—just to name a few.

There are commercially available 'hyperspectral' cameras [10, 17, 18] which capture high-resolution radiance spectrum at each pixel of the scene within a given spectral range (e.g., the 'visible' range that runs roughly from 400 to 700 nanometers). However, those devices often suffer from trade-offs among spectral, spatial and temporal resolutions, while being at a much higher price point compared to the widely available RGB cameras [7].

In RGB imaging, typically 3 types of color sensors are used to capture the light radiances within the visible range, returning 3 intensity values per pixel. Clearly, this common imaging practice is a *lossy* process for the continuous spectral radiance functions. Moreover, different camera manufacturers and/or models use different set of color sensors, which makes the 'R, G, B' a device-dependent measure of light [36]. This constitutes the root of the requirement of ill-posed color conversions/calibrations among color devices, most significantly the color correction process [14] (where different *device-dependently* measured RGBs are mapped to their approximate colorimetric counterparts, e.g., sRGB and XYZ triplets). Spectral Reconstruction (SR) algorithms are another attempt at the standardization of RGB data. Here, a hyperspectral image is inferred from the device RGBs. Of course, with these spectra in-hand we can also solve the color correction problem (by numerical integration).

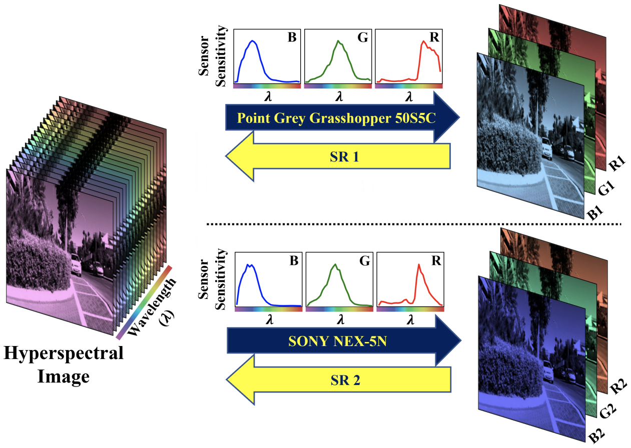

Figure 1. *The RGB imaging outcome depends on the used camera model, and so the reverse process 'Spectral Reconstruction' (SR) should also be dependent of the camera model (i.e., SR 1 $\neq$ SR 2).*

Historically, given matching RGB and hyperspectral measurements, SR algorithms set out to find a statistical mapping from the former to the latter. Early methods include regression [21, 12, 30, 25], Bayesian inference [3, 29, 9] and iterative optimization [8, 41]. Advantageously, these methods only need a small set of point measurements for training. The more recent algorithms, including sparse coding [4, 1, 26] and Deep Neural Networks (DNNs) [6, 7], are trained on hyperspectral 'image' datasets, with a single RGB seen in a spectral and/or spatial neighborhood mapped to a single spectral output. The advantage of this training setup is that hypespectral image datasets provide very large training sets. However, sparse coding (to some extent) and DNN methods are based on orders of magnitudes more parameters to solve for. It is not clear whether even a large hyperspectral image set can serve as a rich enough training set.

The spectral reconstruction experiment is illustrated in Figure 1. Here, RGB images are generated from the hyperspectral ground-truths by numerical integration with the camera sensitivities. In the figure we show that the RGB images could be generated (in this case) from two different cameras—here, Point Grey Grasshopper 50S5C and Sony NEX-5N. Note that their spectral sensitivities are different (see especially the red sensitivity). Then we try and infer the spectral images from the RGBs. Clearly, the mapping function from RGB to spectrum must be different for each camera. Further, it is *a priori* possible that one camera supports better spectral reconstruction than another.

In the prior art, Arad and Ben-Shahar [5] demonstrated that there exists drastic difference in SR performance for clustering-based SR techniques when different cameras are used, Kaya et al. [23] developed a DNN-based method where RGB images from different cameras can be admitted as input, and Fu et al. [15] further proposed a CNN-based model that jointly selects the best camera sensitivities and recover spectra. However, the prior art did not comprehensively investigate the SR performance as a function of both camera and SR method.

In this paper, we present SR results for 10 different camera sensitivity functions (9 real cameras from different brands + CIE 1964 color matching functions [32]), for regression (linear and non-linear), sparse coding, and DNN algorithms. Our study shows three main results. First, for all algorithms the choice of camera makes a small difference in how well spectra can be measured from RGBs. Secondly, for a single camera we observe that the mean performance ranking of the SR methods does not change. Finally, we see that the 'best' camera to use depends on the algorithm at hand (e.g., regression vs DNN) and the difference can be significant.

2. Background

2.1. Color Image Formation

Let us denote $r(\lambda)$ as the light spectrum coming from the scene at one pixel, where λ is the wavelength. Separately, the R, G, and B sensors give weights to each wavelength differently as they integrate the signal $r(\lambda)$ into the respective sensory responses [39]:

$$\int_{\Omega} s_k(\lambda) r(\lambda)\, d\lambda = x_k \ , \quad k = R, G, B \ , \tag{1}$$

where $s_R(\lambda)$, $s_G(\lambda)$ and $s_B(\lambda)$ are called the *spectral sensitivity functions*, Ω represents the *visible range* which approximately runs from $\lambda = 400$ nanometers (nm) to $\lambda = 700$ nm, and x_k is the k-th channel sensory response.

In practice, a hyperspectral imaging device measures $r(\lambda)$ discretely within Ω. Assuming that the samplings are fine enough, we can approximate Equation (1) into a vectorized form:

$$[\underline{s}_R, \underline{s}_G, \underline{s}_B]^{\mathsf{T}} \underline{r} = \underline{x} \ , \tag{2}$$

where $\underline{s}_R$, $\underline{s}_G$, $\underline{s}_B$, and $\underline{r}$ are the n-component discretized vectors of $s_R(\lambda)$, $s_G(\lambda)$, $s_B(\lambda)$, and $r(\lambda)$, respectively, and $\underline{x} = [x_R, x_G, x_B]^{\mathsf{T}}$. Effectively, we are calculating the 'dot products' between the discrete spectral signals (i.e., hyperspectral measurements) and the spectral sensitivity vectors. In general, the spectral sensitivities, $\underline{s}_R$, $\underline{s}_G$, $\underline{s}_B$, are camera specific (fixed for a given RGB camera), and can be measured using a spectrally-scanning monochrometer [22]. Thus, if we are given a set of measured spectral sensitivities of a camera, we can use Equation (2) to simulate the RGB responses of this camera from the hyperspectral measurements.

2.2. Spectral Reconstruction

In Spectral Reconstruction (SR), we wish to infer $\underline{r}$ from $\underline{x}$. Let us denote an SR algorithm as Ψ. In SR we seek to minimize the approximation error of:

$$\Psi(\underline{x}) \approx \underline{r} \ . \tag{3}$$

2.2.1. Linear and Non-Linear Regression

When Ψ is a 'linear' regression [21], a single $n \times 3$ matrix is trained to recover spectra:

$$\Psi(\underline{x}) = \mathbf{M}\underline{x} \ . \tag{4}$$

Polynomial Regression (PR) [12], Root-Polynomial Regression (RPR) [25], and Radial-Basis-Function Network (RBFN) [30], are the non-linear instantiations of Ψ:

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

where φ represents an expansion function. As an example, for second-order RPR, each RGB is expanded to a 6-vector: $[R\ G\ B\ \sqrt{RG}\ \sqrt{BG}\ \sqrt{RB}]^{\mathsf{T}}$, and the $\mathbf{M}$ in Equation (5) is $n \times 6$. Like RPR, the PR and RBFN methods can be thought of as regressions with different expansion functions φ. In this paper, 6-th order polynomial/root-polynomial expansions are used for PR and RPR [12, 25], respectively, and 45 radial basis functions are used for RBFN [30].

Given a set of training data, denoted as set T, the $\mathbf{M}$'s in both Equation (4) and Equation (5) are most commonly solved by least-squares minimization [21]:

$$\min_{\mathbf{M}} \sum_{i \in T} \left(||\Psi(\underline{x}_i) - \underline{r}_i||_2^2 + \gamma ||\mathbf{M}||_F^2 \right) \ , \tag{6}$$

where the SR function Ψ is dependent of $\mathbf{M}$ (and a fixed φ, for non-linear regressions), and γ is an empirical factor that balances the minimization of the fitting error (the former term) and the magnitude of $\mathbf{M}$ (the latter term), which provides stability to the solution against noise in the data [37].

In our experiment, the proper γ for each regression method is determined by a cross-validation method [16]. Specifically, we try different γ values, ranging from 10^{-20} to 10^{20} and finely sampled, and solve for $\mathbf{M}$ iteratively. The γ that returns the minimal mean recovery error over some *validation-set* data (a separate set from the training set T) is selected.

2.2.2. A+ Sparse Coding

The A+ sparse coding method [1] takes a different approach to improving the fitness of linear regression. In A+, it is assumed that a linear regression mapping is sufficient if we separate the RGBs into 'RGB neighborhoods' and consider them separately. Mathematically, we write:

$$neighborhood(\underline{x}) = j \quad \Rightarrow \quad \Psi(\underline{x}) = \mathbf{M}_j \underline{x} \ , \tag{7}$$

where the linear regression matrix $\mathbf{M}_j$ is trained specifically for the RGBs in the j-th neighborhood.

In training, A+ determines the neighborhoods using the K-SVD clustering algorithm [2]. More specificaly, K clusters are derived from the hyperspectral data, whose centers are recorded and projected down to the RGB space using Equation (2). Then, surrounding each of the K RGB centers we find N neighboring RGBs in the training set to train the local linear regression mapping (i.e., the $\mathbf{M}_j$ in Equation (7)). In the inference phase, for each query RGB we simply find its closest RGB cluster center and apply the SR regression matrix attached to the neighborhood.

The number of neighborhoods, K, and the training neighborhood size, N, were determined empirically. In this paper, we simply use the suggested parameters presented in [1], which are $K = 1024$ and $N = 8192$ (out of a reduced training set of 3,000,000 spectra from the ICVL dataset).

2.2.3. HSCNN-R Deep Neural Network

Recently, many SR methods are based on Deep Neural Networks (DNNs). For example, at least 10 new methods were proposed following each of the bi-annual NTIRE Spectral Reconstruction Challenges [6, 7], where most (if not all) methods are based on DNNs. In this paper, we consider HSCNN-R [34]—the second place winner of NTIRE 2018 [6] (and the one we are most familiar with)—as an exemplar DNN method.

Like most of the DNN-based methods, HSCNN-R takes 'patches' of RGB image as inputs, from which both spatial and spectral features are extracted and used to inference spectra. That is:

$$\Psi\big(patch(\underline{x})\big) \approx \underline{r}\ , \tag{8}$$

where $\underline{x}$ and $\underline{r}$ are RGB and hyperspectral measurements at the same pixel, and $patch(\underline{x})$ indicates the RGB image patch centering at this pixel. Specifically, HSCNN-R is based on a deep residual learning framework [20], where multiple 'residual blocks' consisting of 2 convolutional layers and 1 ReLU layer are stacked one after another that constitutes the main feature mapping process of Ψ [34].

Based on one suggestion in [34], the input RGB patch size is set to 50×50, the convolutional kernels are 3×3, the number of filters in each layer is 64, and 16 residual blocks are used (which translates to a depth of 34 layers) in our experiments. Most trainings saturate at around 300 to 350 epochs (the validation-set images are used to determine the saturation of the trainings).

3. Spectral Reconstruction as a Function of Camera and Algorithm

3.1. Dataset

Hyperspectral measurements: We use the ICVL hyperspectral image dataset [4] which comprises 200 scenes of size 1300×1392 as the ground-truth hyperspectral measurements. The spectrum at each pixel is measured at every 10 nanometers (nm) between 400 nm and 700 nm.

Camera Sensitivities: Then, we have the RIT camera sensitivity database of 28 cameras [22], from which we select 9 cameras (from the 9 different brands presented) for our experiments. Additionally, we also use the CIE 1964 color matching functions [32] as a sensor set (not least because they are used in the NTIRE challenges [6, 7]). The 10 selected cameras and their aliases used in this paper are listed in Table 1. All camera sensitivities are measured at (or linearly interpolated to match) the spectral range of hyperspectral measurements (i.e., 10-nm intervals from 400 to 700 nm).

Table 1. 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 [22].

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

Table 2. Considered spectral reconstruction methods.

Abbreviation	SR method
LR	Linear Regression [21]
RPR	Root-Polynomial Regression [25]
A+	A+ Sparse Coding [1]
RBFN	Radial-Basis-Function Network [30]
PR	Polynomial Regression [12]
HSCNN-R	HSCNN-R Deep Neural Network [34]

With both information, we simulate 10 corresponding RGB image sets from the ICVL hyperspectral image set using Equation (2), each with a different set of camera sensitivities.

3.2. Training, Validation, and Testing

The SR methods—introduced in Section 2.2—are summarized in Table 2. As a common practice we split our spectral data into training, validation and testing sets. In our experiment, we randomly select 100 hyperspectral images for training, 50 scenes for validation, and 50 for testing. This training/validation/testing division is fixed for all cameras and all the SR algorithms tested.

To account for how well spectra are recovered from RGBs, we use the Mean-Relative-Absolute Error (MRAE) (the standard metric to rank the methods in NTIRE challenges [6, 7]). The MRAE calculates a percentage recovery error:

$$\mathrm{MRAE} = \frac{1}{n}\left\|\frac{\underline{r} - \Psi(\underline{x})}{\underline{r}}\right\|_1 \times 100\ (\%)\ , \tag{9}$$

where $\underline{r}$ and $\Psi(\underline{x})$ are the ground-truth and reconstructed spectra, n is the length of the spectral vectors ($n = 31$ for the ICVL dataset), and the division is calculated component-wise. For more details about the MRAE, see [27].

3.3. Results

The presented statistics are calculated as follows. First, for each testing-set image, we calculate the MRAE recovery errors for all pixels. Then, the per image mean and 99-percentile errors are calculated. Finally, the mean results over the images (i.e, the 'mean of the means' and the 'mean of the 99-percentile errors') over the testing-set scenes are given in Table 3 and 4, respectively.

For the per-image-mean results, additionally, we conduct a 'paired two-sample Student t-test' [35] for each SR method, to

Table 3. The mean per-image-mean hyperspectral image reconstruction accuracy over the testing image set measured in MRAE. We show MRAE in percentage (%) because it is a relative (percentage) error metric.

	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	6.36	6.31	6.13	6.23	6.09	6.16	6.21	5.96	6.52	6.29
RPR	4.66	5.05	4.99	5.04	4.66	4.96	4.98	5.43	4.96	4.71
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 4. The mean per-image-99-percentile hyperspectral image reconstruction accuracy over the testing image set measured in MRAE (%).

	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	17.47	14.96	14.30	14.69	15.06	14.54	14.68	13.43	16.24	15.37
RPR	16.40	13.88	13.47	13.75	14.45	13.70	13.79	12.69	15.01	14.27
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

Table 5. Student *t*-test results between the mean performances (Table 1) 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	$< 10^{-4}$
RPR	Cam 24	CMF	3.29	$< 10^{-3}$
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	$< 10^{-3}$

see if the mean performance of the 'best camera' is actually better than the 'worst camera' at a statistically significant level. We calculate both the t-test scores and the corresponding p-values for the significance tests. In our case, we have 50 scenes in the test set (i.e., 50 samples of the per-image-mean statistics), and thus the degree of freedom of the test is $50-1=49$. Then, according to the 'one-sided' hypothesis (since we would only like to know if the best camera's mean MRAE is *lower* than the worst camera's), we calculate the p-values (i.e., the level of significance) from the t-test scores given the t-distribution table [24]. Both numbers are given in Table 5. As shown, the performance of the best camera is always statistically better than the worst for all 6 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 2.

3.4. Discussion

The statistics in Tables 3, 4 and 5 show several important results. First of all, let us fix each SR method individually while comparing the performances when different cameras are used (i.e., comparing the numbers horizontally). Clearly, for all methods, both the mean (Table 3) and 99-percentile (Table 4) performances vary when different cameras are used. And, the range of variation (between worst and best case) for mean results is around 9% to 24%, and for 99 percentiles it is in the order of 18 - 30%.

In general, SR methods with lower mean errors also has 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 2. Clearly, while the best cameras lower the general level of errors, some parts of the scene appear to get worse, e.g., the 'tree' part in the top-left corner for A+ and RPR, and the 'sky' part in the top portion for PR and RBFN.

In Table 5, 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 RPR and 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 consistent rankings. The rankings can, occasionally, change for the 99-percentile error. Indeed, we see that for the Olympus E-PL2 camera (Cam 21), RPR delivers a lower 99-percentile MRAE than the A+ algorithm, while the order is reversed for all other 9 cameras.

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 in an article they 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), and if we (wrongly) compare the

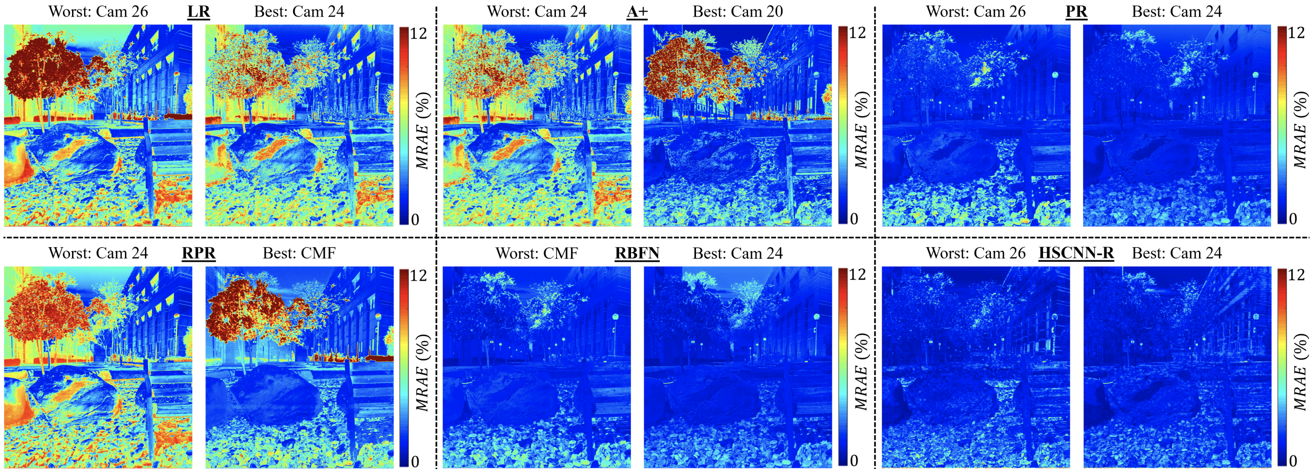

Figure 2. The worst camera vs. best camera comparison for each SR method on one example scene, in terms of the MRAE error heat map.

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 on the same camera (either Cam 24 or Cam 26) for both algorithms, we come to the opposite conclusion (that HSCNN-R is better).

Another aspect to look at this comparison is that, as we 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, i.e., switching the camera used for training PR from Cam 26 to Cam 24 can reach better performance than continuing using Cam 26 and switching to use HSCNN-R for SR (while HSCNN-R is orders of magnitudes more complicated than PR [27]).

4. Conclusion

Spectral Reconstruction (SR) algorithms attempt to recover hyperspectral images from RGB camera responses.In this paper, we investigated the interplay between camera spectral sensitivities and SR algorithms. We chose 10 different sets of camera spectral sensitivities and 6 SR algorithms—including linear and non-linear regressions, sparse coding, and Deep Neural Network (DNN).

Our results allow three broad conclusions to be made. First, for all 6 SR algorithms, there exists a 'best' and a 'worst' camera: the algorithm recovers spectra that are statistically significantly more accurate using the best as opposed to the worst camera. Second, the mean algorithm performance rankings of the tested SR methods do not change for all cameras considered (while the ranking for the worst case 99-percentile error can change slightly). For example, the DNN method we considered is always better than the regression-based methods (e.g., polynomial regression), and this does not change when we change the camera. This said, in contrast we found that, depending on the algorithm, the ranking of the cameras could change markedly. This is a curious result. Indeed, at the margin it means that one algorithm while generally deemed better than others (e.g., DNN is better than polynomial regression) can potentially be overcome simply by switching the used camera (we indeed found that switching to use a different camera to train the polynomial regression can have a greater performance boost than switching to use the DNN for SR while continuing to use the original camera).

References

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

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

[3] N. Akhtar and A. Mian. Hyperspectral recovery from rgb images using gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):100–113, 2018.

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

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

[6] B. Arad, O. Ben-Shahar, R. Timofte, et al. NTIRE 2018 challenge on spectral reconstruction from RGB images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops*, pages 929–938. IEEE, 2018.

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

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

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

[10] Xun Cao, Hao Du, Xin Tong, Qionghai Dai, and Stephen Lin. A prism-mask system for multispectral video acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2423–2435, 2011.

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

[12] D.R. Connah and J.Y. Hardeberg. Spectral recovery using polynomial models. In *Proceedings of the Color Imaging X: Processing, Hardcopy, and Applications*, volume 5667, pages 65–75. International Society for Optics and Photonics, 2005.

[13] L.A. Courtenay, D. González-Aguilera, S. Lagüela, S. Del Pozo, C. Ruiz-Mendez, I. Barbero-García, C. Román-Curto, J. Cañueto, C. Santos-Durán, M.E. Cardeñoso-Álvarez, et al. Hyperspectral imaging and robust statistics in non-melanoma skin cancer analysis. *Biomedical Optics Express*, 12(8):5107–5127, 2021.

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

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

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

[17] N. Gat. Imaging spectroscopy using tunable filters: a review. In *Proceedings of the Wavelet Applications VII*, volume 4056, pages 50–64. International Society for Optics and Photonics, 2000.

[18] R.O. Green, M.L. Eastwood, C.M. Sarture, T.G. Chrien, M. Aronsson, B.J. Chippendale, J.A. Faust, B.E. Pavri, C.J. Chovit, M. Solis, et al. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment*, 65(3):227–248, 1998.

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

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016.

[21] V. Heikkinen, R. Lenz, T. Jetsu, J. Parkkinen, M. Hauta-Kasari, and T. Jääskeläinen. Evaluation and unification of some methods for estimating reflectance spectra from RGB images. *Journal of the Optical Society of America A*, 25(10):2444–2458, 2008.

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

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

[24] S. Kokoska and D. Zwillinger. CRC standard probability and statistics tables and formulae. pages 152–153. CRC Press, 2000.

[25] Y. Lin and G.D. Finlayson. Exposure invariance in spectral reconstruction from RGB images. In *Proceedings of the Color and Imaging Conference*, volume 2019, pages 284–289. Society for Imaging Science and Technology, 2019.

[26] Y. Lin and G.D. Finlayson. Investigating the upper-bound performance of sparse-coding-based spectral reconstruction from rgb images. In *Color and Imaging Conference*, volume 2021, pages 19–24. Society for Imaging Science and Technology, 2021.

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

[28] M. Lv, T. Chen, Y. Yang, T. Tu, N. Zhang, W. Li, and W. Li. Membranous nephropathy classification using microscopic hyperspectral imaging and tensor patch-based discriminative linear regression. *Biomedical Optics Express*, 12(5):2968–2978, 2021.

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

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

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

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

[33] H. Shen and J.H. Xin. Spectral characterization of a color scanner by adaptive estimation. *Journal of the Optical Society of America A*, 21(7):1125–1130, 2004.

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

[35] G.W. Snedecor and W.G. Cochran. Statistical methods 6th edition. pages 91–119. The Iowa State University, Ames, Iowa, 1967.

[36] S. Süsstrunk, R. Buckley, and S. Swen. Standard RGB color spaces. In *Proceedings of Color and Imaging Conference*, volume 1999, pages 127–134. Society for Imaging Science and Technology, 1999.

[37] A.N. Tikhonov, A.V. Goncharsky, V.V. Stepanov, and A.G. Yagola. *Numerical Methods for the Solution of Ill-posed Problems*, volume 328. Springer Science & Business Media, 2013.

[38] O. Torun and S.E. Yuksel. Unsupervised segmentation of lidar fused hyperspectral imagery using pointwise mutual information. *International Journal of Remote Sensing*, 42(17):6461–6476, 2021.

[39] B.A. Wandell. The synthesis and analysis of color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):2–13, 1987.

[40] W. Wang, L. Ma, M. Chen, and Q. Du. 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*, 14:3170–3184, 2021.

[41] S. Zuffi, S. Santini, and R. Schettini. From color sensor space to feasible reflectance spectra. *IEEE Transactions on Signal Processing*, 56(2):518–531, 2008.

Author Biography

Yi-Tun (Ethan) Lin is a Ph.D. student in the Colour & Imaging Lab, University of East Anglia (UEA), UK. He received an M.Sc. degree in Colour Science from the Erasmus+ COSI program in 2018, and a B.Sc. degree in Physics in 2016, from National Taiwan University. His research interest is physics and machine-learning-based spectral reconstruction.

Graham Finlayson is a Professor of Computer Science at UEA, where he leads the Colour & Imaging Lab. Professor Finlayson is interested in 'computing how we see' and his research spans computer science (algorithms), engineering (embedded systems) and psychophysics (visual perception). Significantly, some of Graham's research is implemented and used in commercial products including photo processing software, dedicated image processing hardware and in embedded camera software.