Journal of the Optical Society of America A
OPTICS, IMAGE SCIENCE, AND VISION

Improving RGB illuminant estimation exploiting spectral average radiance

Ilaria Erba,[1,*] Marco Buzzelli,[1] Jean-Baptiste Thomas,[2,3] Jon Yngve Hardeberg,[2] and Raimondo Schettini[1]

[1]Department of Informatics Systems and Communication, University of Milano–Bicocca, 20126 Milan, Italy
[2]Colourlab, NTNU - Norwegian University of Science and Technology, Gjøvik, Norway
[3]ImViA Laboratory, University of Burgundy, Dijon, France
*i.erba3@campus.unimib.it

Received 24 October 2023; revised 25 January 2024; accepted 26 January 2024; posted 26 January 2024; published 22 February 2024

We introduce a method that enhances RGB color constancy accuracy by combining neural network and *k*-means clustering techniques. Our approach stands out from previous works because we combine multispectral and color information together to estimate illuminants. Furthermore, we investigate the combination of the illuminant estimation in the RGB color and in the spectral domains, as a strategy to provide a refined estimation in the RGB color domain. Our investigation can be divided into three main points: (1) identify the spatial resolution for sampling the input image in terms of RGB color and spectral information that brings the highest performance; (2) determine whether it is more effective to predict the illuminant in the spectral or in the RGB color domain, and finally, (3) assuming that the illuminant is in fact predicted in the spectral domain, investigate if it is better to have a loss function defined in the RGB color or spectral domain. Experimental results are carried out on NUS: a standard dataset of multispectral radiance images with an annotated spectral global illuminant. Among the several considered options, the best results are obtained with a model trained to predict the illuminant in the spectral domain using an RGB color loss function. In terms of comparison with the state of the art, this solution improves the recovery angular error metric by 66% compared to the best tested spectral method, and by 41% compared to the best tested RGB method.

https://doi.org/10.1364/JOSAA.510159

1. INTRODUCTION

Color constancy is the human visual system's ability to perceive objects' colors as relatively constant even when the color of the illuminant changes [1]. Human color constancy is achieved through a combination of mechanisms including chromatic adaptation, color contrast, and spatial filtering [2]. Computational color constancy, on the other hand, refers to the process of designing algorithms that enable digital cameras to achieve color constancy, and it is typically addressed as a two-step process, composed of illuminant estimation and illuminant correction [3]. Computational color constancy, from now on referred to as "color constancy" for brevity, is an active area of research in computer vision and digital photography, and many algorithms have been proposed to address this problem. However, achieving human-like color constancy in machines remains a challenging task.

Color constancy is formulated as an inverse problem that aims at reversing the commonly accepted imaging model and separating the reflectance of the object from the illumination:

$$I_k(x, y, \lambda) = \int_\omega L(\lambda) R(x, y, \lambda) S_k(\lambda) \mathrm{d}\lambda, \quad (1)$$

where $R(x, y, \lambda)$ is the surface reflectance, $L(\lambda)$ the illumination property, and $S_k(\lambda)$ the sensor characteristics, as a function of the wavelength λ, over the visible spectrum ω. The subscript k represents the sensor's response in the kth channel and $I_k(x, y, \lambda)$ is the image corresponding to the kth channel ($k = \mathrm{R, G, B}$). Color constancy algorithms must rely on additional assumptions, constraints, or information in order to select a valid solution to estimate the illuminant given the input image I_k. Low-level, statistics-based algorithms make explicit assumptions about the statistical properties of natural scenes, such as the assumption that the color of the light source is typically mostly achromatic. These algorithms estimate the color of the illuminant as the deviation from these assumptions, and they tend to rely on simple statistical operations on the image, such as computing the mean of the color values in the image [4]. In contrast, more recent and effective algorithms are learning based and exploit models trained on handcrafted features extracted from the input image, or deep learning models [5]. These methods make higher-level reasoning about the relationship between image features and illuminant estimation and are expected to rely on assumptions based on the distribution of the training data [6,7]. They tend to be more complex and computationally

intensive but can lead to more accurate and robust illuminant estimation as they can effectively learn and model the complex relationship between image features and illuminant estimation.

Nowadays smartphones may embed spectral sensors that are able to capture the spectral average radiance of the scene. A recent patent from Apple [8], for example, describes an electronic device that includes control circuitry that gathers ambient light measurements using a color ambient light sensor. Sensor responses are processed to generate a color rendering index for the ambient light, which is used to correct the color of the captured images via a color correction matrix. This leads to more accurate and faithful color reproduction in the captured images. Hybrid-resolution spectral imaging systems have also been proposed [9–11], where a conventional high-resolution RGB color camera is combined with a low-resolution spectral imaging sensor, producing a high-resolution spectral image. Our work focuses on investigating how low-resolution spectral radiance can be combined with high-resolution RGB color information to produce a properly white-balanced RGB color image.

Unlike previous works that only consider either the RGB or the spectral information, we combine them both to improve the accuracy of illuminant estimation. Although presented as an investigation, we have developed a method to carry out the work at hand. Our approach involves a neural network that combines spectral and color information through convolutional and feedforward layers. In addition, we have implemented a selection module that uses clustering techniques to extract the best estimation from the one proposed by the network. In this paper we show that, by incorporating both the RGB and spectral domains, we are able to capture a more comprehensive set of features related to the illuminant, which improves the accuracy of the estimation. In particular, we conduct an investigation divided into three points: (1) we want to identify which resolution of color and spectral information brings the higher benefit, (2) we want to investigate whether it is more beneficial to predict the illuminant in the spectral or color domain, and finally, (3) we want to discover if it is better to provide the illuminant target in the color or spectral domain, for the training phase.

The paper is structured as follows: Section 2 introduces the related scientific literature, covering both illuminant estimation in RGB images as well as illuminant/reflectance separation in spectral images. Section 3 describes our proposed method for illuminant estimation exploiting both RGB and spectral average radiance. Section 4 presents the experimental setup and results.

2. RELATED WORKS

Through the years, several methods have been proposed for illuminant estimation in the RGB domain. Statistical methods, such as Grey World [4], White Patch [12], Shades of Grey [13], and Gray Edge [14], use the statistical properties of the scene. While these methods are simple and efficient, they often fail in the presence of non-gray objects in the scene or non-uniform illumination [14,15].

Forsyth [16], and later Gijsenij *et al.* [17], introduced gamut-based methods for color constancy. These are based on the assumption that in real-world images, for a given illuminant, one observes only a limited number of colors. Consequently, any variations in the colors of an image (i.e., colors that are different from the colors that can be observed under a given illuminant) are caused by a deviation in the color of the light source. This limited set of colors that can occur under a given illuminant is called the canonical gamut image. Gamut-based methods have a sensitivity to the scene content similar to that of methods based on lower-level statistics, combined with a non-negligible computational complexity, especially when handling large-resolution images.

More recently, deep-learning-based methods have been proposed for illuminant estimation. Bianco *et al.* [6] proposed a color constancy method using convolutional neural networks (CNNs). They trained a CNN on a large dataset of images with ground truth illuminant color information to estimate the illuminant color from an input image. Hu *et al.* [7] proposed a fully convolutional color constancy method called FC4. Their method uses a fully convolutional neural network to estimate the illuminant color spatial distribution of an input image that is used to correct the input image for color constancy. More recently, alternative approaches for convolution-free deep learning have been applied to illuminant estimation as well: Li *et al.* [18] proposed a transformer-based multiple illuminant color constancy method called TransCC. The method uses a transformer-based network to estimate the illuminant color distributions of an input image under multiple illuminants. The generative nature of the method is what enables the handling of multiple illuminant sources, at the same time however introducing potential artifacts in the output white-balanced images.

These approaches, traditionally applied in RGB imaging, are also at the basis of several works that treat spectral information.

Zheng *et al.* [19] proposed a method that models the separation of illuminant from reflectance as a low-rank matrix factorization task, and developed a scalable algorithm that works in the presence of model error and image noise. They demonstrated that taking advantage of the greater color variety offered by hyperspectral images can improve separation accuracy, and relax the restrictive subspace illumination assumption in the existing literature, thus providing supporting evidence for the method proposed in this work.

Khan *et al.* [20,21] illustrate the potential benefits of using multispectral imaging in computer vision applications, but also acknowledging that multispectral imaging can still be sensitive to changes in illumination. To address this problem, the authors propose directly extending computational color constancy algorithms to multispectral imaging, including edge-based methods [14] as well as highlight-based methods [22]. In subsequent works [23,24] they then developed a spectral adaptation transform to bring the multispectral image data into a canonical representation, effectively performing illuminant correction.

Su *et al.* [25] proposed a separation of the reflectance and illumination components using a weight scheme, factorizing the weighted specular-contaminated pixels to estimate the illumination spectrum. Despite the demonstrated robustness in both simulation and real experiments, it is computationally expensive, since this approach requires a number of iterations for the spectral illuminant estimation of a single image.

Robles-Kelly and Wei [26] proposed using a convolutional neural network to estimate the illuminant in spectral images. The network takes an input tensor constructed from an image

patch at different scales, which allows it to predict the illuminant per-pixel using locally supported multiscale information.

More recently, Kitanovski *et al.* [27] developed an imaging pipeline for a spectral filter array camera to estimate scene reflectances in the absence of knowledge about the scene illuminant. The proposed approach involves estimating the illuminant's spectral power radiance, which is shown to stabilize and marginally improve the estimation accuracy compared to the method that estimates the illuminant in the RGB domain only.

Li *et al.* [28] formulated the problem of multispectral illumination estimation as a matrix factorization task, and used the alternating direction method of multipliers optimization algorithm to solve it, by unrolling it into a multi-stage network.

Unlike more traditional computational color constancy algorithms that estimate illuminants from RGB images, Koskinen *et al.* [29] propose to use the average color spectra of a scene. They tested several regression functions (such as kernel ridge, random forest, and multilayer perceptron) to map the spectral pixel to the white point. They demonstrate that the method is effective even with as few as 10–14 spectral channels.

3. METHOD

Our work poses itself with the purpose of providing a color illuminant estimation method that combines RGB color and spectral information. Assuming the availability of an RGB color image and the spectral average distribution of its corresponding radiance scene, the combination is performed by means of a suitably designed neural network. According to our method, we divide the input image into patches, and we train our designed neural network having as input the RGB color image patch, and its corresponding spectral average distribution. The size of the patch may vary, and its tuning is discussed in Section 4.B. The process for each single patch is visually depicted in Fig. 1, where we illustrate in parallel the process of RGB color illuminant estimation and spectral illuminant estimation. These two options will be compared in Section 4.B. Given an input image, the individual patch estimations are combined with a suitable selection module, as described in Section 3.A.

The neural network architecture depicted in Fig. 2 is composed of two branches. The first branch takes as input the RGB color patch having size $w \times h \times 3$, and a suitably designed convolutional neural network (CNN) extracts a feature vector of size N, where N is the same as the spectral resolution. The second branch takes as input an N-dimensional vector of the spectral average distribution, and a feed-forward neural network (FFNN) extracts a feature vector of the same size N. The two vectors are then concatenated into a vector of size $2N$, which is fed to the final block of the neural network, which differs in structure and in terms of the final output:

- an encoder for the case of RGB color illuminant estimation produces as output a three-dimensional vector corresponding to the RGB coordinates of the illuminant;
- a feed-forward neural network (FFNN) for the case of spectral illuminant estimation produces an N-dimensional vector corresponding to the spectrum of the illuminant.

The choice of the convolutional part of the network architecture takes into consideration the scarce availability in the state of the art of spectral datasets that provide illuminant targets of spectral radiance images. Due to this circumstance, we opted for a shallow convolutional neural network architecture with a small number of trainable parameters. This implementation choice is carried out additionally to the fact that the images are divided into smaller patches, which already increases the number of input images, both for the training and testing phases. More precisely, we selected the convolutional mean architecture [30], which consists of two convolutional layers (the first being $3 \times 3 \times 3 \times 7$ and the second one $3 \times 3 \times 7 \times 14$, both of them having stride and padding set to one) each followed by a max pooling layer (2×2) and the activation function, next the weighted global average pooling layer, which in turn is composed by the third convolutional layer ($1 \times 1 \times 14 \times 3$ with stride set to one and padding set to zero) followed by a ReLu and a per-channel global average pooling. While the per-channel global average pooling layer returns the feature map averaged by a channel, we only reduce by half the feature map dimension with an average pooling layer and then we feed the resulting feature map to a fully connected layer to return a vector equal in size to the number of channels of the spectral input. The purpose of using a 1×1 block is to apply weights to each output feature channel after Conv2 and obtain a three-channel output, in the case of the convmean network, and a 31-channel output in ours. We replaced ReLu layers with leaky ReLu [31], which has proven to be capable of solving the "dead neuron" problem and

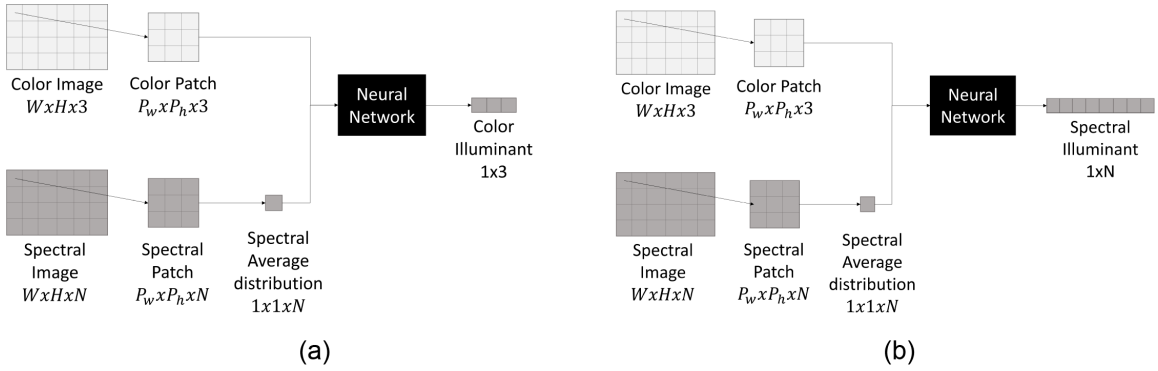

Fig. 1. RGB color and N-dimensional spectral images are divided into patches of $P_w \times P_h$ pixels. The RGB color patch and the spectral average distribution are fed to the neural network. In (a) the network returns an RGB color illuminant estimation while in (b) it returns a spectral illuminant estimation.

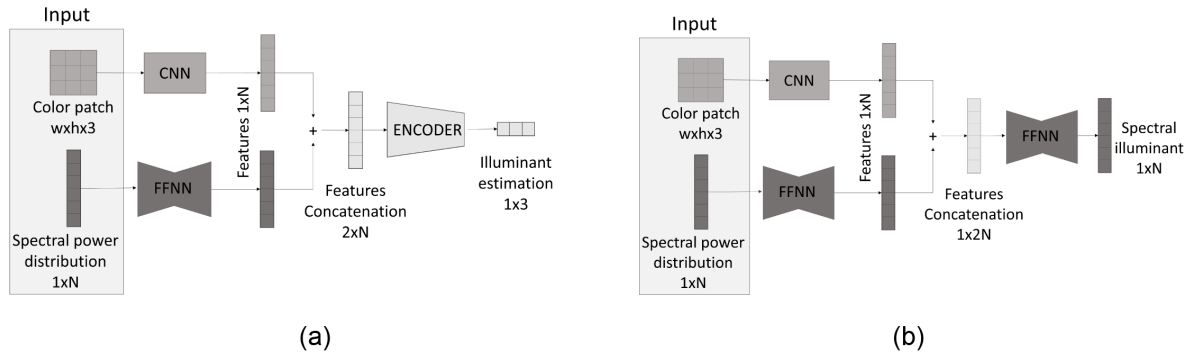

Fig. 2. (a) Color architecture; (b) spectral architecture. The two architectures are identical except for the last block. The network gets in input a color patch (of dimension $h \times w \times 3$) and its spectral average distribution. The color patch is fed to the convolutional neural network, which returns a $1 \times N$ feature map. In the same way, the spectral average distribution is fed to a feed forward neural network that elaborates it and returns a $1 \times N$ feature map. The two feature maps get concatenated into a unique feature map. In (a) the resulting feature map gets fed to another feed forward neural network that finally returns a $1 \times N$ spectral illuminant estimation. In (b), instead, the resulting future map gets fed to an encoder, which returns the color illuminant estimation.

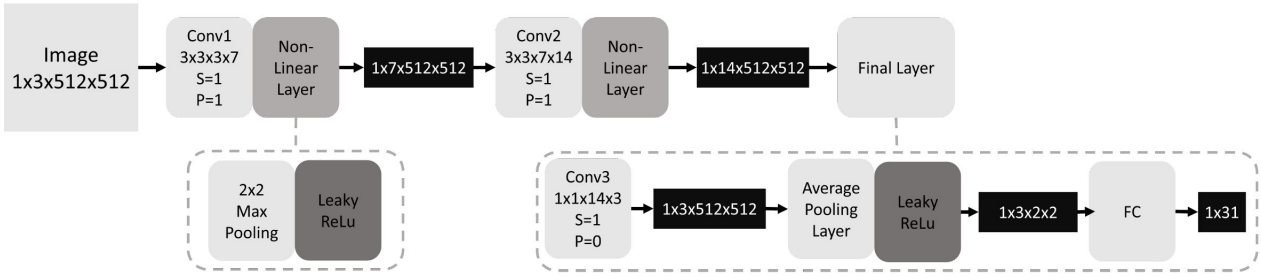

Fig. 3. Convolutional neural network block architecture contains two 3×3 filter convolutional layers (Conv1/2), which are followed by a 2×2 max pooling and a leaky ReLU. In the end, there is a final layer that is implemented as a 1×1 convolutional layer (Conv3) with leaky ReLU, an average pooling layer that reduces the feature map dimension, and finally a fully connected layer (FC). In this diagram, P and S denote padding and stride, respectively. The other four numbers shown in the Conv box represent "Filter Size 1 × Filter Size 2 × #Input Channel × #Output Channel" whose product is the total number of filter parameters. The activation function is displayed assuming a patch size of 512.

is more effective than ReLu. Although leaky ReLu can return negative values, we chose it because it facilitates gradient back-propagation during the initial stages of training. As the training progresses, the network learns to estimate positive outputs from the provided ground truth data. The diagram of the CNN block is shown in Fig. 3.

In Fig. 4 we show the FFNN structures identified in dark gray in Fig. 2. This consists of three fully connected layers followed by the leaky ReLu activation function. The first and second layers of the first structure map the N-band spectral input (or feature map) to 60 values, while the first and second layers of the second structure map a $2N$-band spectral vector to 60 values. Finally, the last layer in both structures maps them back to N values.

In Fig. 5 we show the encoder architecture that produces the RGB illuminant estimation for an input patch. It consists of three fully connected layers: the first one maps the N input values to 15 values, the second one maps them to five, and the final layer maps them to three, obtaining, therefore, a color

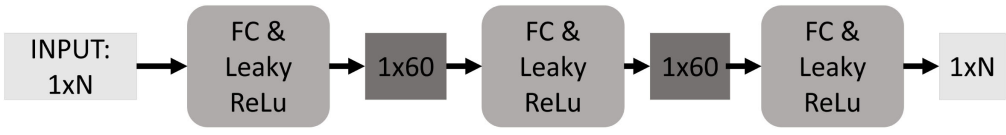

Fig. 4. Feed forward neural network block architecture consists of three hidden layers. The rectangles with round edges indicate the layers used; in this specific case "FC" stands for fully connected layers, which are then followed by a leaky ReLu activation function.

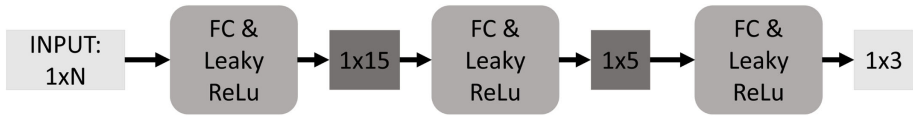

Fig. 5. Encoder block architecture consists of three hidden layers. As for Fig. 4, the rectangles with round edges indicate the fully connected layers, which are again followed by a leaky ReLu activation function.

illuminant estimation. For the activation function, we select the leaky ReLu, for the same reasons previously explained.

A. Selection Module

Our model provides several illuminant estimations corresponding to the patches of a single input image. Assuming that the illumination is uniform across the scene, we exploit a further module in the inference phase with the purpose of selecting the best color estimation among the several color estimates corresponding to the analyzed patches. Although the network's illuminant estimations can be fused to solve the multi-illuminant estimation problem, the main focus is on a global illuminant. The intuition behind this module is that, among the suggested estimations, some are to be considered outliers, which we want to eliminate, leaving us with those estimations that are supposedly closer to the actual illuminant in the scene. In order to implement this intuition, we exploit a k-means clustering [32] to assess a consensus among the estimations. We carry out this process in the inference phase, after the multispectral-RGB conversion step; therefore, the clustering is computed in the RGB color domain. The number of clusters is automatically determined by computing the silhouette coefficient [33]. The populousness of the clusters determines which are to be considered outliers and which one is to be taken into consideration to determine the most likely solution. We propose two alternative selection strategies to determine such solution: the cluster centroid, and the individual patch estimation that is closest to the cluster centroid. These two strategies are compared in Section 4.B.

4. EXPERIMENTS

This work focuses on the development of a neural network able to estimate the RGB color of the illuminant of a scene by combining spectral and RGB color information. To this end, the image is divided into patches, and for each patch we used the average radiance and the RGB data of the patch itself. In the inference phase the patches' color estimations are further processed to produce a single illuminant estimation in the case of using of our selection module.

In our experiments, not only do we want to investigate whether the average spectral distribution of the scene can improve the RGB illuminant estimation, but we also want to investigate the influence of the input patch size. This section also revolves around another turning point: whether predicting the illuminant in the RGB color or spectral domain is more valuable as input for RGB color correction. The second option, which consists of predicting a spectral illuminant, brings up two training strategies: (1) training the spectral prediction with the spectral expected illuminant or (2) converting the prediction to RGB color and then training it with the RGB color expected illuminant (ground truth). To this end, a straightforward solution for the multispectral-to-raw conversion consists of integrating the multispectral illuminant estimation with the camera sensitivity functions. This section will show and analyze the results obtained from these three "implementation choices." From now on we will refer to them as:

1. RGB color architecture (CA);
2. spectral architecture trained on spectral (SATOS);
3. spectral architecture trained on RGB color (SATOC).

The performance of the model is evaluated through the recovery angular error metric [34] defined as follows:

$$e_{\text{rec}}(U, V) = \arccos\left(\frac{U \cdot V}{|U||V|}\right), \tag{2}$$

where "$\cdot$" indicates the dot product, $|x|$ is the euclidean norm, U denotes the RGB illuminant target, and V is the RGB estimated illuminant.

A. Dataset

This work requires a dataset that contains both spectral and RGB color images in full resolution and the corresponding target illuminant in both representations. Although multiple datasets in the state of the art provide both spectral images and the ground truth illuminant, we limit our experiments to the NUS dataset since it is the only one that acquires images in real-world scenarios. The NUS dataset [35] contains 64 spectral radiance images, of which 24 are reserved for testing and 42 for training. The images have dimensions of $1312 \times W \times 31$ pixels, where W varies from 951 to 2374. For each spectral image, a total of 31 bands were captured at 400–700 nm, with a spacing of 10 nm. The scenes' subjects include outdoor and indoor images and natural and man-made objects. For illumination sources, the dataset varies from natural sunlight and shade conditions, additionally considering artificial wide-band lights obtained from metal halide lamps of different color temperatures (2500 K, 3000 K, 3500 K, 4300 K, 6500 K) and a commercial off-the-shelf LED E400. Furthermore, the dataset is provided with the spectral radiance of the scenes and the camera sensitivity function, which allows for spectral to color conversion. The target illuminant is retrieved from color-checker targets present in the spectral radiance images. The dataset is provided under the assumption that the illuminant is global over the entire scene.

According to our method, the input RGB and spectral images are divided into patches. Given the unequal width dimensions of the dataset elements, to prove the effectiveness of our approach we decided to limit our analysis to a 512×512 central crop of the image. These images are further divided into patches of sizes: 4×4, 8×8, 16×16, 32×32, 64×64, 128×128, 256×256, and finally 512×512. The RGB color patches and the corresponding average spectral radiance are used to train our model, as shown in Fig. 2. The performances for the different patch sizes are discussed in the next sections.

B. Experimental Results

The investigation hereby conducted mainly focuses on the resolution of color and spectral information, and which combination of these two domains best benefits the color illuminant estimation problem.

1. Patch Illuminant Estimation

Given the 24 test images, we will assess the network's capability to accurately estimate the illuminant of the single image

Table 1. Performance of the Three Methods Measured in Terms of Recovery Error[a]

Method	Patch Size	Min	Mean	Median	ple95	Max
CA	4	0.03	1.92	0.68	9.37	26.85
	8	0.03	**1.71**	0.37	7.16	15.41
	16	0.03	1.93	**0.18**	7.52	16.96
	32	0.03	1.80	0.20	7.58	17.25
	64	0.03	1.81	0.55	6.72	15.91
	128	0.03	2.04	1.12	6.34	15.53
	256	0.06	1.98	1.09	6.36	15.96
	512	0.08	2.52	1.94	**4.90**	**13.86**
SATOS	4	0.03	2.34	0.32	9.44	39.72
	8	0.03	2.10	0.47	8.24	29.14
	16	0.03	2.29	**0.21**	12.16	17.26
	32	0.03	**1.93**	0.22	9.17	15.50
	64	0.03	1.97	0.31	10.20	15.52
	128	0.03	1.84	1.06	7.65	15.02
	256	0.03	2.30	1.65	**6.81**	16.74
	512	0.76	2.90	1.79	6.92	**12.31**
SATOC	4	0.03	1.90	**0.04**	10.07	45.25
	8	0.03	2.21	0.10	9.45	21.22
	16	0.03	1.91	0.13	8.20	17.39
	32	0.03	1.84	0.37	7.60	15.46
	64	0.03	**1.67**	0.73	6.56	14.07
	128	0.03	1.77	0.80	6.42	15.40
	256	0.03	1.80	0.83	7.09	16.26
	512	0.69	2.80	1.87	**6.40**	**11.80**

[a]The recovery error is in degrees (the lower, the better). In bold we highlighted the best results based on the method.

patches. The estimated illuminants are compared with the target illuminants (i.e., the illuminant associated to the image), and the resulting recovery angular errors across all patches from all test images are synthesized in Table 1 with several statistics: min, mean, median, percentile 95, and max. As explained in Section 3, we proposed two different architectures (one that estimates the illuminant in the RGB color domain, and one in the spectral domain) and two training strategies (one provides a color target illuminant, while the other provides a spectral target illuminant). For the sake of simplicity, we sum up these training strategies as (1) color architecture (trained on a color target) (CA), (2) spectral architecture trained on a spectral target (SATOS), and (3) spectral architecture trained on a color target (SATOC).

As we can see in Table 1 the mean recovery error values go from 1.67° to 2.9°, respectively achieved by the SATOC on patch size 64 and the SATOS for patch size 512. We can notice from the metrics illustrated in Table 1, and even more evidently in Fig. 6, that spectral and color resolution greatly impact the performance of the methods. The patch of 512 × 512, which has the highest color resolution and the lowest spectral resolution, performs the worst. At the same time, though, the best-achieving patch sizes are not the ones with the highest spectral resolution (from 4 × 4 to 16 × 16), but the mid-size ones (from 32 × 32 to 128 × 128). In Fig. 6 the continuous lines represent the mean recovery angular error values for different patch sizes. The colored areas indicate the 95% confidence interval, which means that there is a 95% probability that the estimation for a given patch size falls within that range. It was expected that one of the methods would perform better than the other two for each patch size. However, the average values indicate that there is no clear winner between CA and SATOC, while SATOS is the least effective option except for patch sizes of 8 × 8 and 128 × 128. Method CA, instead, performs best for patch sizes 8 × 8 and 512 × 512, while from patch sizes 64 × 64 to 256 × 256 the SATOC method performs the best; the performance for the remaining patch sizes is very similar. The confidence interval gets smaller as the patch size decreases; however, it is important to highlight that the smaller patch sizes have a very large number of estimations, making it easier for the estimation to fall within a restricted interval.

The median values analysis confirms the results obtained with the mean values. The patch size 512 × 512 is the worst-performing patch, while the best-performing patch size for both CA and SATOS is 16 × 16. For the SATOC method, the best-performing patch size is 4 × 4. It is also peculiar that patch size 512 × 512 is the best-performing patch size for the max metric. However, we do not consider it to be statistically relevant. As the 512 × 512 patch size corresponds to the entire image, it contains more information for estimation. Patch size 4 has the highest error rate, which means it is easier for the method to receive an unhelpful patch for that particular estimation. For example, the global illumination assumption might be incorrect, and a patch with a different illuminant from the one measured through the color checker may be used. Moreover, a white patch may be more useful than a black patch.

To help the reader in understanding the results of patch illuminant estimation using the SATOC method, we have included an example image in Fig. 7. This image is divided into patches of varying sizes, and each patch is color-coded based on its recovery angular error value. The figure also provides a visual representation of the selection module process.

2. *Image Illuminant Estimation*

According to the proposed method, we have several candidates for illuminant correction in a given image, corresponding to the

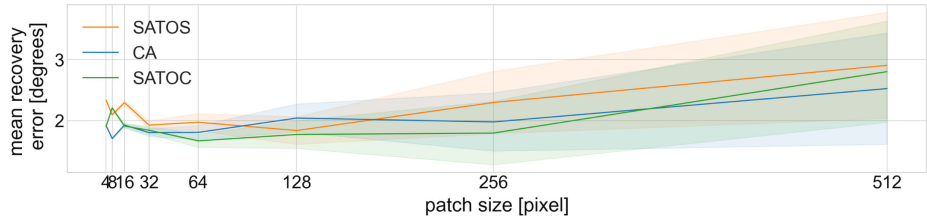

Fig. 6. Accuracy of the CA, SATOS, and SATOC methods, in terms of mean recovery angular error in degrees (°) varying the patch size with their confidence interval. The lower the result, the better.

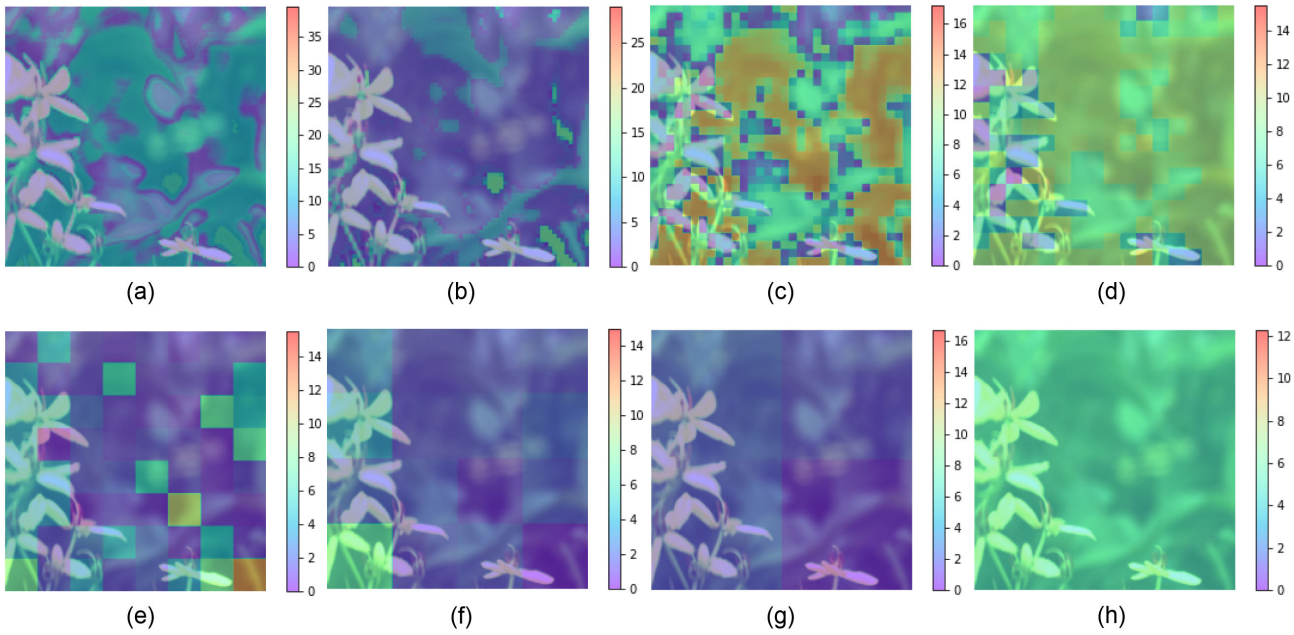

Fig. 7. Visual demonstration of the SATOC illuminant estimation method for a given image. We have overlaid a heatmap on the image to indicate the recovery angular value for each patch. The patches with lower errors appear bluish, while those with higher errors appear reddish. (a) Patch size 4, (b) patch size 8, (c) patch size 16, (d) patch size 32, (e) patch size 64, (f) patch size 128, (g) patch size 256, and (h) patch size 512.

image patches. For this reason, we provided the method of an additional selection module, which has the purpose of identifying a single illuminant estimation among the ones estimated from the patches. As explained in Section 3.A we achieve this by relying on a simple clustering technique.

In Table 2 the results of the proposed selection module are shown, based on multiple criteria. First, we investigate whether the centroid of the most populated cluster is the one closest to the ground truth. The assumption behind the selection module is that the majority of estimations proposed by the network are close to the target illuminant. The clustering process serves as a voting mechanism; therefore, the cluster containing the majority of the estimations also supposedly contains the estimation closest to the ground truth. Therefore, we included an accuracy metric to investigate the veracity of this assumption: the metric is defined as the number of selected centroids that actually contain the estimation closest to the ground truth, normalized by the number of images N and expressed as a percentage. We can see from Table 2 that the selection cluster accuracy goes from 44% to 88%. Overall the assumption is verified in more than 60% of the cases.

We then evaluate the performance of the two different proposed illuminant selections: centroid and estimation closest to the centroid. As also shown in both Figs. 8 and 9, the performances of the two selection strategies are almost identical. From the graph, it is easy to see that the CA model overall has the best performance, except for patch sizes 4, 8, and 32. The best performance overall is achieved by the CA model for patch size 8 with a recovery error of 1.33°. The graph also confirms that the SATOS model is the worst-performing among the proposed ones. Figure 9 shows the performance of the selection module compared to the patch illuminant estimation average performance. The results show that the selection module improves the average performance of the method for all the training strategies.

Table 2. Selection Module Performance[a]

Method	Patch Size	Selection Cluster Accuracy	Centroid Recovery Error	Closest to Centroid Recovery Error
CA	256	64%	1.82	1.82
	128	56%	1.98	2.00
	64	76%	1.57	1.58
	32	72%	1.53	1.52
	16	64%	1.76	1.75
	8	**88%**	**1.33**	**1.33**
	4	80%	1.58	1.58
SATOC	256	76%	1.78	1.80
	128	56%	1.83	1.83
	64	68%	1.56	1.57
	32	72%	1.75	1.74
	16	**80%**	**1.46**	**1.46**
	8	76%	1.90	1.90
	4	**80%**	1.63	1.64
SATOS	256	44%	2.28	2.22
	128	48%	1.78	1.76
	64	60%	**1.62**	**1.63**
	32	64%	1.89	1.89
	16	64%	2.31	2.31
	8	**68%**	1.86	1.86
	4	64%	2.04	2.05

[a]The selection cluster accuracy is expressed in percentage, where 100% is the best result. Centroid recovery error and closest to centroid recovery error are expressed in degrees; the lower, the better. In bold we highlight the best results based on the selection method.

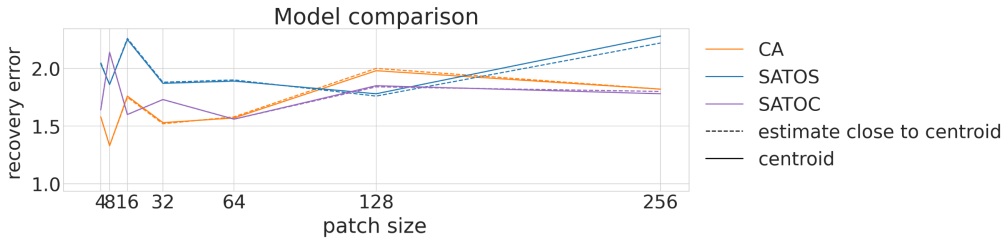

Fig. 8. Comparison between the centroid and the estimation closest to the centroid performance. The comparison is performed in terms of recovery error in degrees (°); the lower, the better.

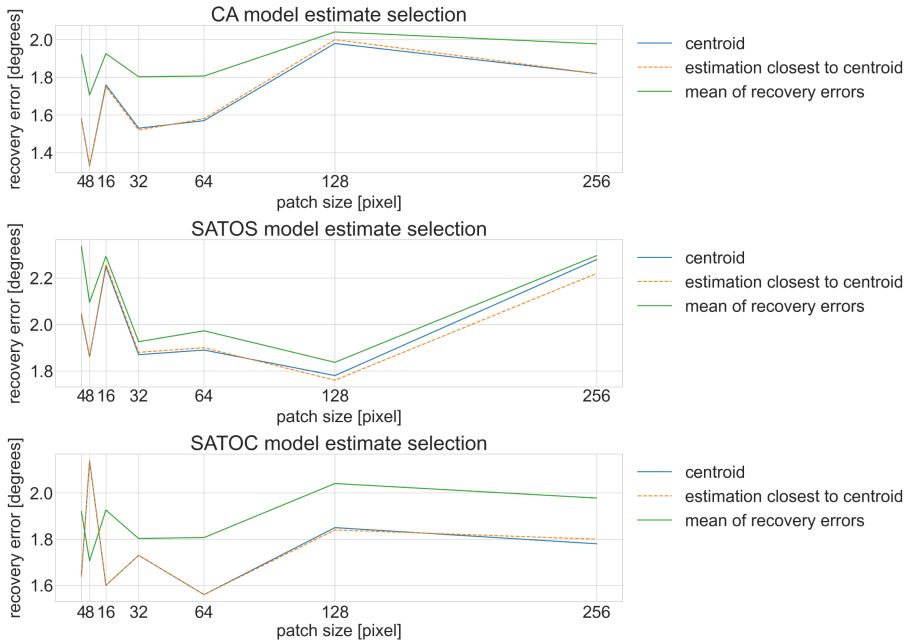

Fig. 9. Comparison of performance for the three models (CA, SATOS, and SATOC) with centroid selection, estimation closest to the centroid, and performance before the estimate selection. The comparison is performed in terms of recovery error (in °; the lower, the better) based on the patch size.

We present some visual examples of the testing images in Fig. 10 after they have been color-corrected using the illuminant estimations obtained with the selection module. The images are accompanied by their respective recovery error values for better interpretation. We also included the original images and their corrected versions using the ground truth for visual comparison. The visual examples confirm the results presented in the previous table. SATOS is the least-performing model among the three, and the performance decreases as the patch size increases.

C. State of the Art Comparison

In this section, we aim to compare the performance of our method with state-of-the-art methods. Our research examines whether the combination of spectral and color information is advantageous in solving the RGB illuminant estimation problem. To achieve this, we will compare our method with both spectral-based and RGB-based methods. For spectral-based methods, we have selected Robles-Kelly's [26] and Khan's [24] methods, which have been extensively discussed in Section 2. As the focus of this work is on spectral-based methods, we will introduce RGB-based methods in this section, but they are beyond the scope of this research.

We offer a range of AWB methods, which include both traditional solutions based on handcrafted features, and newer data-driven approaches based on deep learning. All the methods we analyze are sensor-independent. If training is required, we make use of official pre-trained models to ensure optimal conditions.

Van de Weijer [14] proposed a framework in 2007 to generalize multiple algorithms based on low-level image statistics. By adjusting parameters, several known algorithms can be derived. For this study, six different automatic white balance algorithms were selected by varying the configurations of parameters (Minkowski norm p and standard deviation σ):

- Grey World (GW): $p = 1$, $\sigma = 0$;
- White Point (WP): $p = \infty$, $\sigma = 0$;
- Shades of Grey (SoG): $p = 4$, $\sigma = 0$;
- General Grey World (GGW): $p = 9$, $\sigma = 9$;
- First Order Grey Edge (GE1): $p = 1$, $\sigma = 6$;
- Second Order Grey Edge (GE2): $p = 1$, $\sigma = 1$.

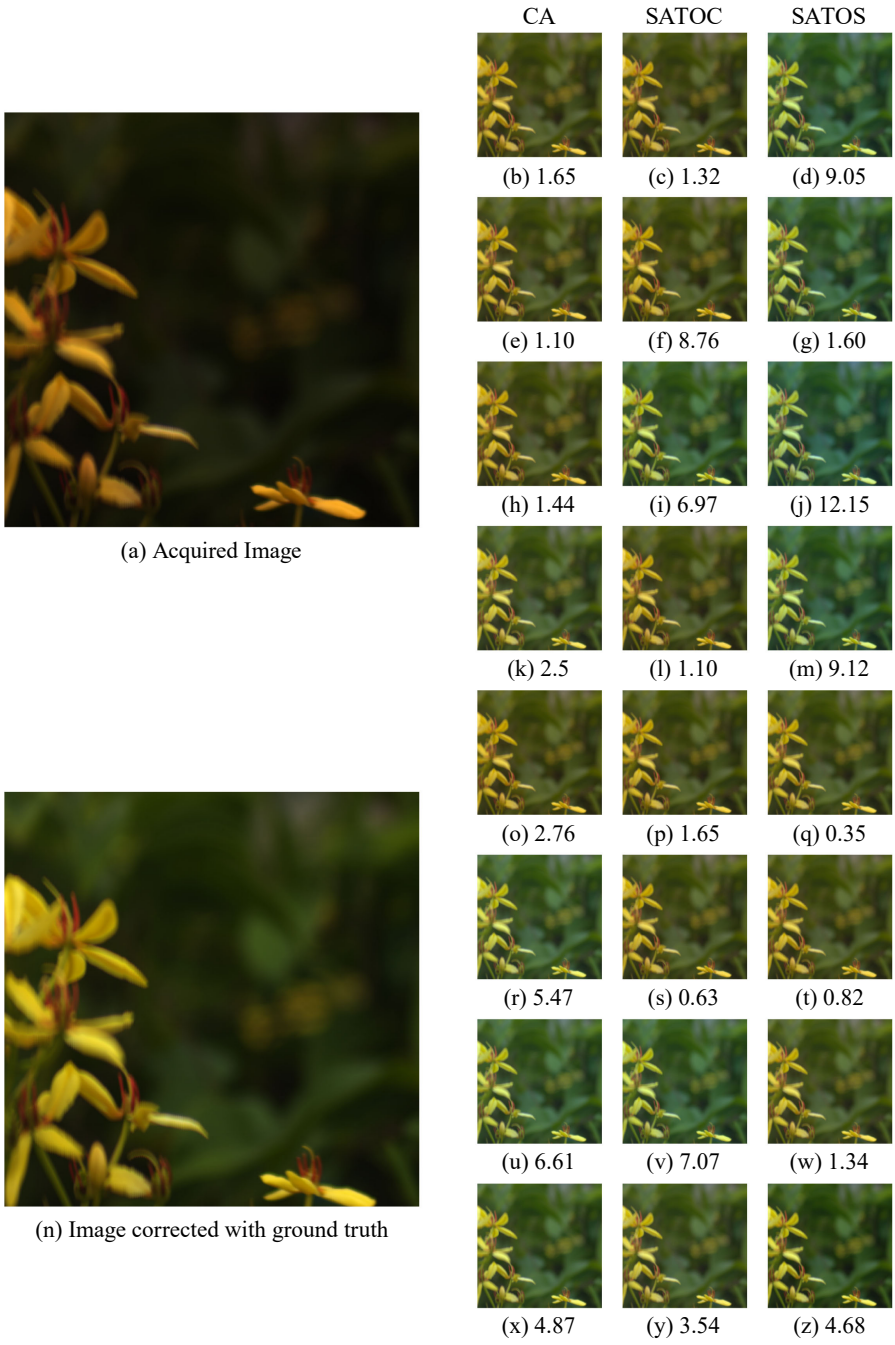

Fig. 10. Images on the right are obtained by applying the illuminant estimation for the color correction. We report the recovery error in degrees in the caption below the images. The images are arranged in three columns, each representing a different model (CA, SATOC, SATOS), and eight rows, each representing a different patch size ranging from 4 to 512.

Among the handcrafted methods we have selected the work of Qian *et al.* [36]. This work proposed a learning-free method called grayness index (GI), which can be used to identify neutral surfaces in an image. They used the dichromatic reflection model [37] to estimate single and multiple illuminants. We have used the default parameters in our implementation, as suggested by the authors.

We chose four data-driven methods for our comparison.

Afifi and Brown [38], instead, developed a learnable sensor-independent pseudo-RAW space to map the RGB values of any given camera, under the explicit assumption of input linear RAW-RGB images. The method is called sensor-independent illumination estimation (SIIE).

Akbarinia and Parraga [39] introduced adaptive surround modulation (ASM), a method that models visual neurons using two overlapping asymmetric Gaussian kernels and weighs their contributions based on center-surround contrast. We used the default parameters provided by the authors.

In their paper, Cheng *et al.* [40] proposed an auto white balance (AWB) algorithm that uses principal component analysis

Table 3. Mean Recovery Angular Error on NUS Dataset [35][a]

Method	Mean Recovery Angular Error	% Improvement Mean Recovery Error
Our baseline (CA_8)	**1.33**	\
Robles-Kelly *et al.* [26]	12.56	89%
Khan *et al.* [24]	3.96	66%
First Order Grey-Edge (GE1) [14]	5.46	76%
Second Order Grey-Edge (GE2) [14]	5.55	76%
General Grey World (GGW) [14]	3.67	64%
Grey World (GW) [14]	3.81	65%
Shades of Grey (SoG) [14]	3.84	65%
White Point (WP) [14]	4.81	72%
Grayness index (GI) [36]	2.41	45%
PCA [40]	3.12	58%
Quasi-unsupervised (QU) [41]	2.25	41%
SIIE [38]	6.19	79%
Adaptive surround modulation (ASM) [39]	3.69	64%

[a]The mean recovery angular error is in degrees (the lower, the better). For our method we selected the CA_8, being the best-performing one. In bold the best results.

(PCA). The method involves selecting a certain percentage of dark and bright pixels for the calculation. The authors achieved the best results with a percentage parameter of 3.5%, which we have adopted as well.

Convolutional neural networks (CNNs) have proven to be effective in various applications. In one instance, Bianco and Cusano [41] created a CNN-based quasi-unsupervised color constancy (QU) algorithm that identified achromatic pixels in color images. The network was trained without explicit AWB annotation. The only assumption made was that the images were roughly balanced.

To make the comparison process easier, Table 3 is divided into three sections. The first row displays the performance of the best model presented in this work. We selected it by choosing the model with the lowest mean recovery angular error value. The second section presents the spectral-based methods, and the third section shows the RGB-based methods. Our best-performing solution (CA on image patches of size 8×8 pixels) was compared to the works of Khan and Robles-Kelly [24,26]. The results show that our method outperforms Khan's and Robles-Kelly's work, respectively, by 66% and 89% for the mean recovery angular error metric. The comparison with the RGB-based methods shows that our method still performs better than the selected methods; more precisely, our method improves the performance for the RGB illuminant estimation problem from 41% for the quasi-unsupervised method to 79% for the SIIE method.

1. Considerations and Observations

Our work shows that the patch size that leads to the best performance is patch size 8×8, and overall, the mid-size patches are the most suited for the problem, indicating that the problem benefits the most from mid-resolution both for color and spectral information. The network that provides estimations in the spectral domain and receives the target in the spectral domain (SATOS) is the worst-performing one. However, no precise pattern was identified in the results, namely, no method seems to perform clearly better for each patch size. Figure 8 and Table 2 show that the CA approach shows a tendency to perform better with smaller patch sizes (4, 8, 32, except for 16) and SATOC shows a tendency to perform better for the larger patch sizes (64, 128, 256, except for 16). We also proved that the selection module performs better than the average of the result, meaning that it can extract an illuminant estimation closer to the target illuminant most of the time. We also show that the potential for improvement is very large; in fact, the error of the models with patch size 4×4 is close to zero. This result also proves how spectral information may be beneficial for the color illuminant estimation problem.

5. CONCLUSION

Spectral sensors are becoming every day cheaper and more available on the market, so much so that they are making their first appearances in digital imaging acquisition tools. This work poses itself as an investigation option to verify if the combination of spectral and RGB color information can improve the result for the RGB color constancy problem.

Our investigation has been performed on the standard NUS dataset. The best results, as identified in this experimental setup, are obtained with a model trained to predict the illuminant in the spectral domain using an RGB color loss function. During the investigation, we focused on three main points. Firstly, our observation was that processing data in the form of mid-size image patches generally yields better results compared to using the whole image or smaller patches. However, the investigation results for the second point were not conclusive as both CA and SATOC models performed similarly well. Lastly, we found that estimating and training in the multispectral domain are not effective for estimating RGB illuminants.

We further evaluated our method against other solutions from the state of the art. To ensure a fair comparison, we selected methods that were trained and optimized on the same experimental setup, as well as sensor-independent methods. Our method demonstrated superior performance within this experimental setup, in terms of RGB illuminant estimation. These results should be further confirmed by future experiments, extending the investigation to a larger properly annotated dataset. Nonetheless, our experiments show in practice the potential of combining spectral and RGB color information to improve RGB illuminant estimation.

Future developments may focus on neural network architectures that not only provide an illuminant estimation but also a level of confidence, as well as spatially varying multi-illuminant estimation.

Funding. Ministero dell'Università e della Ricerca.

Acknowledgment. This work was partially supported by the MUR under the grant "Dipartimenti di Eccellenza 2023-2027" of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca, Italy.

Disclosures. The authors declare no conflicts of interest.

Data availability. The work and results presented in this paper are based on the original NUS dataset by Nguyen *et al.* [35].

Supplemental document. See Supplement 1 for supporting content.

REFERENCES

1. P. V. Gehler, C. Rother, A. Blake, *et al.*, "Bayesian color constancy revisited," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2008), pp. 1–8.
2. A. Hurlbert and K. Wolf, "Color contrast: a contributory mechanism to color constancy," Prog. Brain Res. **144**, 145–160 (2004).
3. D. Cheng, A. Abdelhamed, B. Price, *et al.*, "Two illuminant estimation and user correction preference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 469–477.
4. G. Buchsbaum, "A spatial processor model for object colour perception," J. Franklin Inst. **310**, 1–26 (1980).
5. V. Agarwal, B. R. Abidi, A. Koschan, *et al.*, "An overview of color constancy algorithms," J. Pattern Recogn. Res. **1**, 42–54 (2006).
6. S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks," IEEE Trans. Image Process. **26**, 4347–4362 (2017).
7. Y. Hu, B. Wang, and S. Lin, "Fc4: fully convolutional color constancy with confidence-weighted pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 4085–4094.
8. P.-C. Hung and Z. Zhang, "Electronic devices with color compensation," U.S. patent 11,308,846 (19 April 2022).
9. Y. Murakami, M. Yamaguchi, and N. Ohyama, "Hybrid-resolution multispectral imaging using color filter array," Opt. Express **20**, 7173–7183 (2012).
10. Y. Murakami, K. Nakazaki, and M. Yamaguchi, "Hybrid-resolution spectral video system using low-resolution spectral sensor," Opt. Express **22**, 20311–20325 (2014).
11. K. Nakazaki, Y. Murakami, and M. Yamaguchi, "Hybrid-resolution spectral imaging system using adaptive regression-based reconstruction," in *Image and Signal Processing: 6th International Conference, ICISP 2014, Cherbourg, France, June 30–July 2, 2014*, (Springer, 2014), pp. 142–150.
12. E. H. Land and J. J. McCann, "Lightness and retinex theory," J. Opt. Soc. Am. **61**, 1–11 (1971).
13. G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Color and Imaging Conference* (Society for Imaging Science and Technology, 2004), pp. 37–41.
14. J. Van De Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," IEEE Trans. Image Process. **16**, 2207–2214 (2007).
15. H. R. V. Joze and M. S. Drew, "Exemplar-based color constancy and multiple illumination," IEEE Trans. Pattern Anal. Mach. Intell. **36**, 860–873 (2013).
16. D. A. Forsyth, "A novel algorithm for color constancy," Int. J. Comput. Vis. **5**, 5–35 (1990).
17. A. Gijsenij, T. Gevers, and J. Van De Weijer, "Computational color constancy: survey and experiments," IEEE Trans. Image Process. **20**, 2475–2489 (2011).
18. S. Li, J. Wang, M. S. Brown, *et al.*, "Transcc: transformer-based multiple illuminant color constancy using multitask learning," (2022).
19. Y. Zheng, I. Sato, and Y. Sato, "Illumination and reflectance spectra separation of a hyperspectral image meets low-rank matrix factorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1779–1787.
20. H. A. Khan, J.-B. Thomas, J. Y. Hardeberg, *et al.*, "Illuminant estimation in multispectral imaging," J. Opt. Soc. Am. A **34**, 1085–1098 (2017).
21. H. A. Khan, J.-B. Thomas, and J. Y. Hardeberg, "Towards highlight based illuminant estimation in multispectral images," in *Image and Signal Processing: 8th International Conference, ICISP*, Cherbourg, France, 2–4 July 2018 (Springer, 2018), pp. 517–525.
22. A. Gijsenij, T. Gevers, and J. Van De Weijer, "Improving color constancy by photometric edge weighting," IEEE Trans. Pattern Anal. Mach. Intell. **34**, 918–929 (2011).
23. H. A. Khan, J. B. Thomas, and J. Y. Hardeberg, "Multispectral constancy based on spectral adaptation transform," in *Image Analysis: 20th Scandinavian Conference, SCIA*, Tromsø, Norway, 12–14 June 2017 (Springer, 2017), pp. 459–470.
24. H. A. Khan, J.-B. Thomas, J. Y. Hardeberg, *et al.*, "Spectral adaptation transform for multispectral constancy," J. Imaging Sci. Technol. **62**, 20504 (2018).
25. T. Su, Y. Zhou, Y. Yu, *et al.*, "Illumination separation of non-Lambertian scenes from a single hyperspectral image," Opt. Express **26**, 26167–26178 (2018).
26. A. Robles-Kelly and R. Wei, "A convolutional neural network for pixelwise illuminant recovery in colour and spectral images," in *24th International Conference on Pattern Recognition (ICPR)* (IEEE, 2018), pp. 109–114.
27. V. Kitanovski, J. B. Thomas, and J. Y. Hardeberg, "Reflectance estimation from snapshot multispectral images captured under unknown illumination," in *Color and Imaging Conference (CIC)* (Society for Imaging Science and Technology, 2021), pp. 264–269.
28. Y. Li, Q. Fu, and W. Heidrich, "Multispectral illumination estimation using deep unrolling network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 2672–2681.
29. S. Koskinen, E. Acar, and J.-K. Kämäräinen, "Single pixel spectral color constancy," in *The 32nd British Machine Vision Conference* (2021).
30. H. Gong, "Convolutional mean: a simple convolutional neural network for illuminant estimation," (2020).
31. J. Xu, Z. Li, B. Du, *et al.*, "Reluplex made more practical: leaky relu," in *IEEE Symposium on Computers and communications (ISCC)* (IEEE, 2020), pp. 1–7.
32. J. Yadav and M. Sharma, "A review of *k*-mean algorithm," Int. J. Eng. Trends Technol. **4**, 2972–2976 (2013).
33. H. B. Zhou and J. T. Gao, "Automatic method for determining cluster number based on silhouette coefficient," Adv. Mater. Res. **951**, 227–230 (2014).
34. S. D. Hordley and G. D. Finlayson, "Reevaluation of color constancy algorithm performance," J. Opt. Soc. Am. A **23**, 1008–1020 (2006).
35. R. M. Nguyen, D. K. Prasad, and M. S. Brown, "Training-based spectral reconstruction from a single rgb image," in *European Conference on Computer Vision* (Springer, 2014), pp. 186–201.
36. Y. Qian, J.-K. Kamarainen, J. Nikkanen, *et al.*, "On finding gray pixels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 8062–8070.
37. S. A. Shafer, "Using color to separate reflection components," Color Res. Appl. **10**, 210–218 (1985).
38. M. Afifi and M. S. Brown, "Sensor-independent illumination estimation for DNN models," arXiv, arXiv:1912.06888 (2019).
39. A. Akbarinia and C. A. Parraga, "Colour constancy beyond the classical receptive field," IEEE Trans. Pattern Anal. Mach. Intell. **40**, 2081–2094 (2017).
40. D. Cheng, D. K. Prasad, and M. S. Brown, "Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution," J. Opt. Soc. Am. A **31**, 1049–1058 (2014).
41. S. Bianco and C. Cusano, "Quasi-unsupervised color constancy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 12212–12221.