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

Color constancy from a pure color view

SHUWEI YUE AND MINCHEN WEI*

Color, Imaging, and Illumination Laboratory, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China
**minchen.wei@polyu.edu.hk*

Received 9 December 2022; revised 15 January 2023; accepted 31 January 2023; posted 31 January 2023; published 27 February 2023

Great efforts have been made on illuminant estimation in both academia and industry, leading to the development of various statistical- and learning-based methods. Little attention, however, has been given to images that are dominated by a single color (i.e., pure color images), though they are not trivial to smartphone cameras. In this study, a pure color image dataset, "PolyU Pure Color," was developed. A lightweight feature-based multilayer perceptron (MLP) neural network model—"Pure Color Constancy (PCC)"—was also developed for estimating the illuminant of pure color images using four color features (i.e., the chromaticities of the maximal, mean, brightest, and darkest pixels) of an image. The proposed PCC method was found to have significantly better performance for pure color images in the PolyU Pure Color dataset and comparable performance for normal images in two existing image datasets, in comparison to the various state-of-the-art learning-based methods, with a good cross-sensor performance. Such good performance was achieved with a much smaller number of parameters (i.e., around 400) and a very short processing time (i.e., around 0.25 ms) for an image using an unoptimized Python package. This makes the proposed method possible for practical deployments.

https://doi.org/10.1364/JOSAA.482698

1. INTRODUCTION

Computational color constancy aims to adjust the colors in an image to how they would appear under a canonical illuminant, which is achieved by removing the color cast of the illuminant. For modern digital cameras (e.g., smartphone cameras), it is equivalent to auto white balance and considered as a critical step in image signal processing pipeline to affect the image quality. Great efforts have been made to develop methods for estimating the illuminant of the captured images, leading to the development of various statistical- and learning-based methods [1]. A comprehensive review of the various methods is out of the scope of this article, and only the representative methods are discussed here.

Conventional statistical-based methods generally make some assumptions about the statistical characteristics of the illuminant and/or the colors in a scene, such as white patch [2], gray world [3], shades of gray [4], bright [5], and PCA-based [6] methods, since estimating the illuminant of a scene is actually an ill-posed problem. For example, the gray world method assumes that the average color of all the pixels in an image should be neutral. These statistical-based methods can be implemented efficiently in practice, but their assumptions are not always satisfied in real scenes. Therefore, they commonly result in inaccurate illuminant estimations and, thus, poor image quality. For instance, when a captured scene is dominated by a single color (i.e., a pure color image), the assumptions made by the above methods would be violated.

Recently, deep neural network (DNN) has been adopted in the development of learning-based methods, which has been found to have good performance initially in Bianco *et al.* [7]. An image is commonly divided into many regions, with the estimated illuminant of each region used to estimate the illuminant of the entire image. Hu *et al.* [8], however, found that the illuminants estimated from ambiguous regions, which are dominated by a single pure color, could be significantly different from those estimated from the regions containing semantic information since a given set of camera rgb values could suggest either a surface having such rgb values under a white illuminant or a white surface under an illuminant with such rgb values.

Similar conditions could also happen to an entire image, with the entire image containing an individual color (i.e., a pure color image). This makes the image lack semantic information for illuminant estimation. Such images indeed are common in daily life, with some examples shown in Fig. 1(a). The recent development of smartphone cameras allows close-up (i.e., macro) and telephoto shots, making pure color images appear more frequently, with some examples shown in Fig. 1(b). No past work, however, has investigated the performance of various existing methods for such pure color images, and none of these methods were developed by considering the characteristics of such pure color images. More importantly, no pure color image dataset is available, which could be the reason for the lack of research.

With the above in mind, this article introduces a pure color image dataset—"PolyU Pure Color"—with Fig. 1(a) showing some examples, and also a lightweight feature-based DNN method—"Pure Color Constancy (PCC)" method—for estimating the illuminant of pure color images. This method was found to significantly improve the accuracy of illuminant estimation for pure color images, as well as other normal images,

Corrected 3 March 2023

(a) (b)

Fig. 1. Examples of pure color images (i.e., an image that is dominated by a single pure color): (a) example images contained in the "PolyU Pure Color" dataset collected by us (note: the images are shown with a gamma correction for better visualization); (b) example close-up (i.e., macro) and telephoto shots captured using smartphones.

in comparison to the current state-of-the-art methods. More importantly, this method only contains around 400 parameters and takes around 0.25 ms to process an image in an unoptimized Python version, which makes it possible for implementation in practice.

2. "PolyU PURE COLOR" IMAGE DATASET

A. Existing Datasets for Illumination Estimation

There are several image datasets that are widely used in various research, such as the ColorChecker [9], NUS-8 [6], Cube+ [10], and TAU datasets [11], but none of them were purposely collected to include pure color images. The ColorChecker dataset consists of 568 high-resolution RAW images captured by two cameras (i.e., Canon 1D and Canon 5D). The original images were processed by Shi and Funt to 16-bit linear TIFF images [12]. The dataset was later updated to the Recommend Color Checker (CC2018) dataset with better ground-truth illuminants [13]. The NUS-8 dataset contains 1736 images captured by eight cameras, with only around 210 images captured by each camera, which is considered relatively small for developing learning-based methods [10]. The Cube+ dataset contains 1707 images captured by one camera. The distribution of the ground-truth illuminants of these 1707 images is generally similar to that of the images in the NUS-8 dataset. This can probably explain why a method trained on the NUS-8 was found to have better performance on the Cube+ dataset, which will be discussed in detail in Section 5. The TAU dataset contains 7022 images captured by three cameras, which is considered the largest dataset for illumination estimation.

B. "PolyU Pure Color" Dataset

The PolyU Pure Color image dataset was purposely collected to contain pure color images, including outdoor scenes (e.g., green grass, blue sky, and flowers) and indoor scenes (e.g., illuminated fabrics and walls). Currently, the dataset contains 102 RAW images (i.e., DNG file) that were captured using a Huawei P50 Pro smartphone camera with automatic settings. The original images were 12 bits, with a resolution of 4096×3072, in a BGGR pattern, with a black level of 256 for each channel. The images were processed with the black level subtracted, the two G values averaged, the resolution downsampled to 2048×1536, and saved in a PNG format.

When capturing each image, another image, with an X-Rite ColorChecker placed in the scene, was captured, allowing the collection of the ground-truth illuminant in the scene. The ground-truth illuminant was then calculated using the rgb values of the 20th patch in the color checker (i.e., the second patch in the last row), with the noise reduced using a filter, and the normalized rgb values of the 24 patches are saved in the JSON file so that the images can be processed directly without masking the color checkers. The images and the rgb values of the 24 patches can also be used for data augmentation [8,14,15]. Both the images with and without the color checker were also saved in a JPG format, which was processed by the smartphone automatically for visualization. Figure 2 shows the example of one image set, including the RAW images with and without the color checker and the JPG images for visualization. The thumbnails of the images included in the dataset are shown in Appendix A.

Though the number of images contained in the dataset is relatively small, in comparison to other existing datasets (e.g., the

Fig. 2. Example of an image set included in the "PolyU Pure Color" dataset. Top row, RAW images; bottom row, JPG images; left column, images without the color checker; right column, images with the color checker.

Cube+ dataset), the images were purposely captured to be dominated by a single color, and the color checker information is available for data augmentation.

3. PROPOSED "PURE COLOR CONSTANCY" METHOD

A. Related Work

Many past works have suggested the effectiveness of estimating illuminants based on important features. The Convolutional Color Constancy (CCC) method [16] considers a 2D log-chromaticity histogram as a critical feature, translating the illuminant estimation problem to a classification problem. Then the FFCC method [17], which performs the convolutional operation with a fast Fourier transform (FFT) instead of a Gaussian pyramid, was proposed. Though the FFCC method is significantly faster, it does not always have a higher accuracy than the CCC method [18]. In addition, both the CCC and FFCC methods need the generation of a histogram of each image, which is time consuming and requires great computational power, in terms of floating point operations (FLOPs). Moreover, these two methods treat the illuminant estimation as a classification problem, with the estimated illuminants selected from the image histogram chromaticities (i.e., the log UV chromaticities), which causes the performance to be significantly affected by the size of the histogram.

On the other hand, a greater number of feature-based methods adopt regression methods. For example, Finlayson [19] used the results of the gray world method (i.e., the mean RGB values of the training images) as the inputs for the iterative least squares regression method, leading to much better performance than many learning-based methods and suggesting the effectiveness of using simple features for illuminant estimation. The method using the regression tree proposed by Cheng *et al.* [20] uses four color features (i.e., average chromaticities, brightest chromaticities, dominant color chromaticities, and the chromaticity mode of the color palette) as the inputs. The latter two features require the generation of histograms for each image, which is costly for both processing time and resources, with a model size of 31.5 MB.

B. "Pure Color Constancy" Method

Our proposed PCC method is inspired by the above feature-based methods, especially the method proposed by Cheng *et al.* [20]. Our method, however, aims to perform illuminant estimation based on important color features without requiring complicated pre-processing. For a captured scene, the normalized maximal, mean, brightest, and darkest pixel values are likely to vary under different illuminants, as illustrated in Fig. 3(a), which are considered as four important features for estimating the illuminant. Moreover, based on the images in the PolyU Pure Color Image dataset, we found that these four features tend to form a cluster that can be easily identified, as illustrated in Fig. 3(b).

Thus, the PCC method was designed to use four important color features, in terms of the normalized chromaticities $\{r, g\} = \{R, G\}/(R + G + B)$, as the inputs, which are intensity invariant [10]. These four color

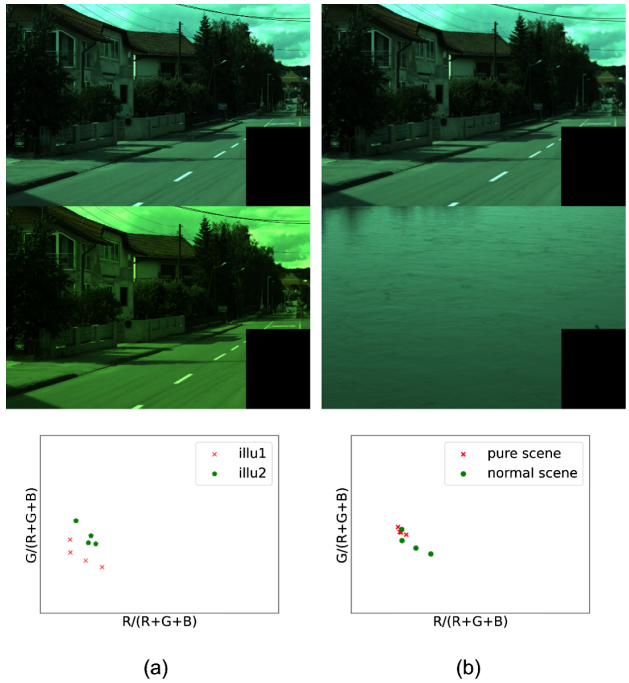

Fig. 3. Illustration of the four important color features—normalized maximal, mean, brightest, and darkest chromaticities—of different captured scenes. (a) Identical scene under two illuminants, with the four features varying with the illuminants; (b) normal scene versus a pure color scene, with the four features clustered together for the pure color scene, suggesting the effectiveness in identifying a pure color image based on these four features.

features are as follows: (1) **max chromaticities**: the chromaticities of the maximal RGB values in each channel (2) **mean chromaticities**: the chromaticities of the mean RGB values in each channel; (3) **brightest chromaticities**: the chromaticities of the pixel having the largest $R + G + B$ value; and (4) **darkest chromaticities**: the chromaticities of the pixel having the smallest $R + G + B$ value.

These four sets of chromaticities are used as the inputs for a lightweight multilayer perceptron (MLP) neural network, which was inspired by [21]. As shown in Fig. 4, the network only contains five hidden layers, with each layer only containing eight neurons and a corresponding ReLU operation. In total, there are around 400 parameters for this shallow network. The output of the network is a set of estimated chromaticities $(\hat{r}, \hat{g})$ for the illuminant in the 2D chromaticity color space, with $\hat{b}$ calculated as $1 - \hat{r} - \hat{g}$. The proposed PCC method only takes 0.25 ms to process an image, based on an unoptimized Python version, which is around 10 times faster than the FFCC method (Model Q), which takes around 2.37 ms for each image [17], and around 100 times faster than the FC4 method, which takes around 25 ms for each image [8]. Moreover, the number of the parameters of the proposed PCC is around 20 times smaller than that of the FFCC method (8200 parameters) and around 10,000 times smaller than that of the FC4 method (4,340,000 parameters).

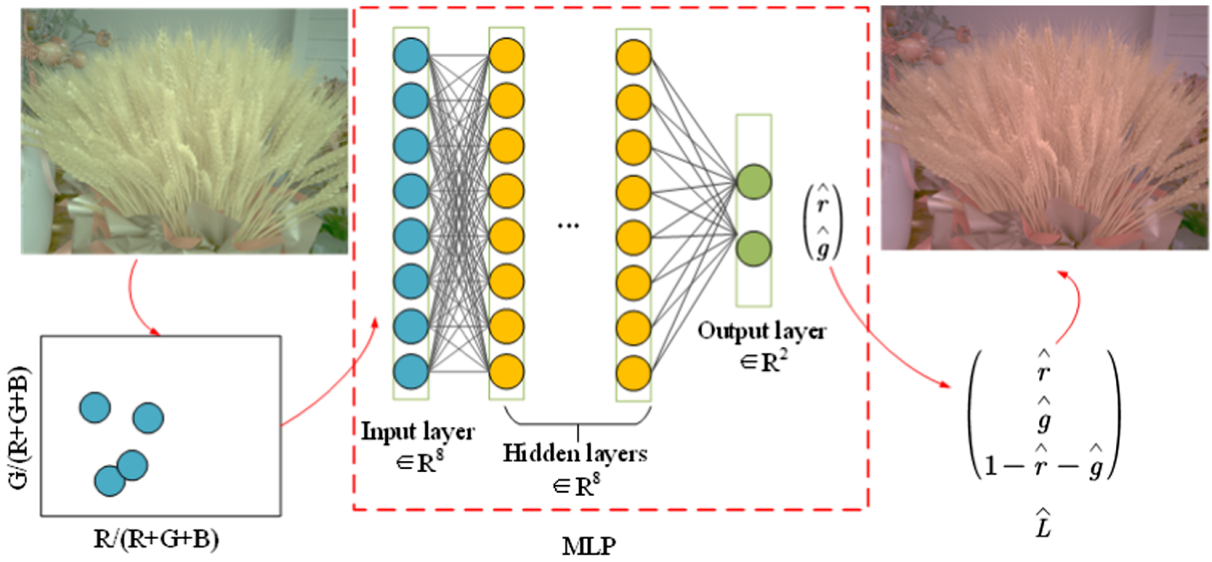

Fig. 4. Overview of the proposed PCC method, with the four color features, in terms of the normalized chromaticities, as the inputs for a lightweight multilayer perceptron (MLP) neural network. It predicts a set of 2D $(\hat{r}, \hat{g})$ chromaticities as the estimated illuminant $\hat{L}$.

4. EXPERIMENT AND RESULTS

A. Settings

The network was trained in PyTorch [22], and Adam [23] was adopted as the optimization algorithm, with a learning rate of 10^{-3} and a cosine annealing schedule [24]. The batch size was set to 1, and the model with the best performance through a total of 10,000 or 100,000 epochs was saved (note: the 10,000 epochs for the CC2018 dataset, and the 100,000 epochs for the other datasets). The standard angular error between the estimated and ground-truth illuminants, as calculated using Eq. (1), was adopted as the error and loss function,

$$\text{error} = \frac{180}{\pi} \arccos\left(\frac{\hat{L} \cdot L}{\|\hat{L}\| \cdot \|L\|}\right), \tag{1}$$

where $\hat{L}$ is the estimated illuminant and L is the ground-truth illuminant.

B. Data Augmentation and Pre-Processing

There are various methods for performing data augmentation for color images. For example, small images that are cropped from a high-resolution image can be used as the training set, with the uncropped regions used as the testing set [18]. On the other hand, some other data augmentation methods adopt a full-matrix (Full-Aug) [14,15] or a diagonal-matrix (AWB-Aug) [8] to transform the original images, which shares a similar concept as relighting.

We adopted the AWB-Aug method for data augmentation, multiplying a 3×3 diagonal matrix M, with the diagonal elements of $[r_a/r_o, g_a/g_o, b_a/b_o]$, to an original image I_0 to derive an augmented image I_a (i.e., $I_a = I_0 \times M$). Though this was adopted in [8], the data augmentation was performed by randomly assigning the RGB values between 0.6 and 1.4, which could lead to illuminants that do not exist in reality. Therefore, we carefully selected a series of illuminants from all the real illuminants of the captured images in the datasets. Then the data augmentation was performed by randomly assigning the RGB values with the chromaticity distance to the selected illuminants smaller than 0.01, as illustrated in Fig. 5.

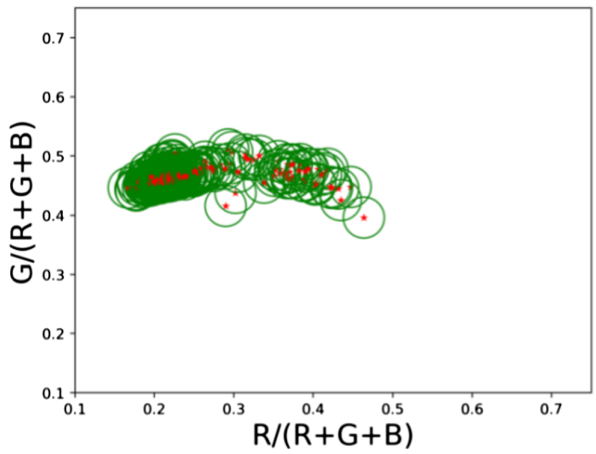

Fig. 5. Illustration of the data augmentation. The red dots are the chromaticities of a series of illuminants that were carefully selected from the illuminants of the captured images in the dataset. The data augmentation was performed by randomly assigning the RGB values with the chromaticities within the green circle, with the chromaticity distance below 0.01 to the selected illuminants.

The raw images were resized to a resolution of 64×64, with a batch of the resized images fed to the network for training and testing. The dark and saturated pixels, with the r, g, or b value below 0.02 or beyond 0.98, were removed. The standard three-fold validation method with a fixed seed was used in the experiment.

C. Results

1. "PolyU Pure Color" Dataset

The performance of the proposed method on pure color images in the PolyU Pure Color dataset was compared with various statistical-based methods—white patch (WP), grey

Table 1. Summary of the Performance of Various Statistical- and Learning-Based Methods, and the Proposed PCC Method, on the PolyU Pure Color Dataset, in Terms of the Angular Errors between the Estimated and Ground-Truth Illuminants[a]

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
WP	9.64	8.78	8.93	2.78	17.25
GW	10.37	8.14	8.67	2.60	21.40
SoG ($p = 3$)	9.66	8.03	8.13	2.38	19.92
Bright ($p = 5$)	8.96	6.92	7.24	2.01	19.57
PCA ($p = 3.5$)	8.32	6.72	6.81	1.74	18.4
FC4 (Squeeze Net)	4.42	3.81	4.05	1.27	8.38
FFCC (Model Q)	3.56	1.97	3.05	0.58	9.39
PCC	3.23	1.40	1.86	0.48	8.76

[a]The listed methods were reevaluated with the parameters shown in the parentheses.

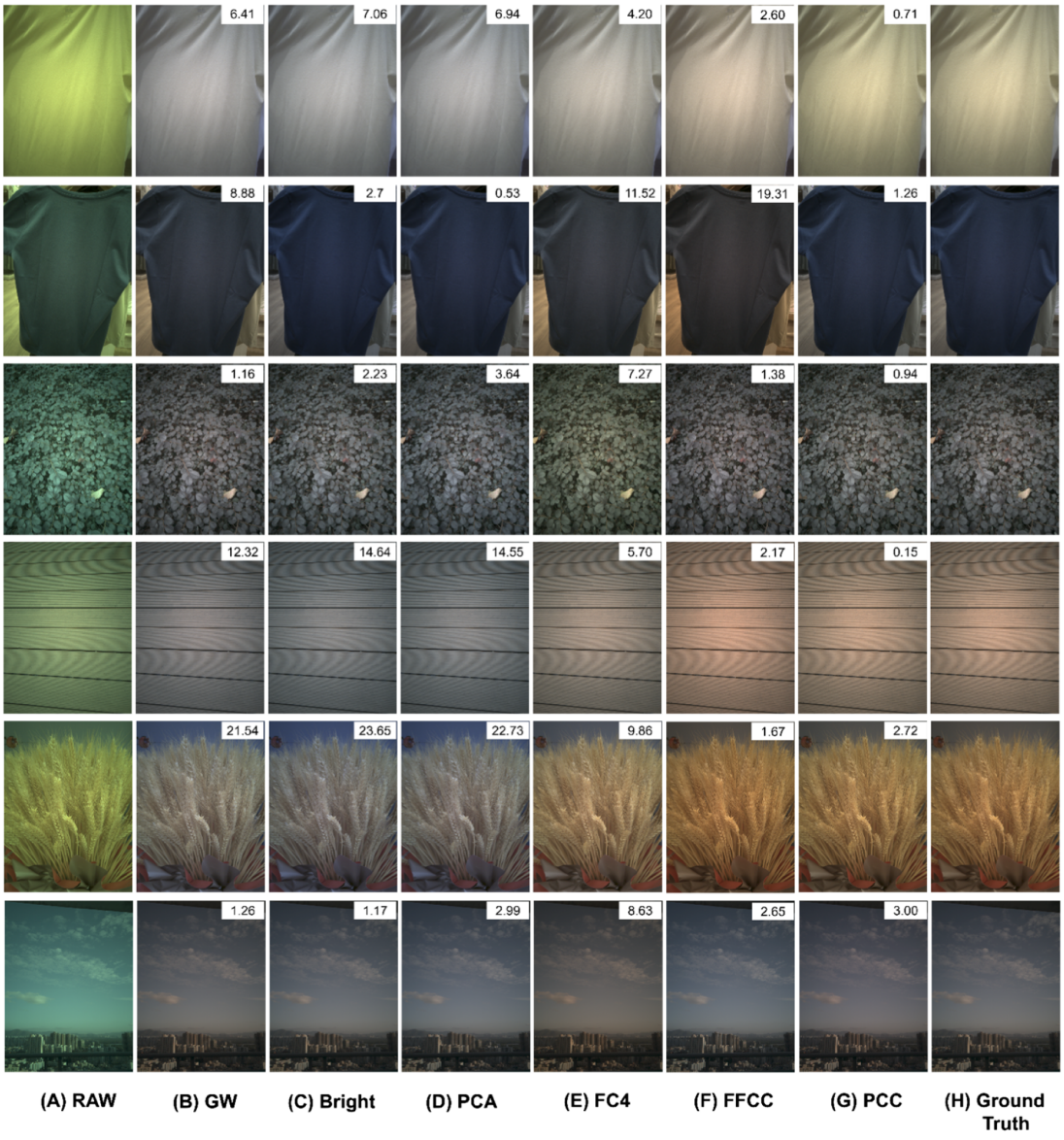

Fig. 6. Examples of the images in the PolyU Pure Color dataset. Column (A) shows the raw image, columns (B) to (F) show the images that were white balanced using the estimated illuminants derived using the existing methods, column (G) shows the image that was white balanced using the estimated illuminant derived using the proposed PCC method, and column (H) shows the image that was white balanced using the ground-truth illuminant. The angular error between the estimated and ground-truth illuminants is shown in each image (note: all the images are shown with a gamma correction for better visualization).

world (GW), shades of gray (SoG), bright pixels (Bright), and Cheng's PCA method (PCA)—and various state-of-the-art learning-based methods—FFCC and FC4, which were retrained according to the literature. Table 1 summarizes and compares the performance based on the angular error between the estimated and ground-truth illuminants, in terms of the mean, median (Med), trimean (Tri), best 25%, and worst 25%, with some example images shown in Fig. 6.

Table 1 clearly shows that the FFCC method resulted in the smallest angular errors among all the existing methods, and the proposed PCC method outperformed all the existing methods, with the angular errors significantly reduced. Though the FFCC method resulted in the smallest angular errors for some individual images, as shown in Fig. 6, the proposed PCC method generally had better performance with little variation on average.

2. CC2018 and NUS-8 Datasets

In addition to pure color images, it is also worthwhile to investigate how the proposed PCC method works for normal images, such as those contained in the CC2018 and NUS-8 datasets. Besides the various existing methods described above, some additional methods—corrected moment with 19 edges (CM), regression tree (RT), CCC, and CNN—were also included. Table 2 summarizes the angular errors derived using the various methods on these two datasets, together with the number of parameters for each method listed in the last column. For the NUS-8 dataset, the proposed PCC model was trained and evaluated on each of the eight cameras, and the results were calculated using the mean results of all cameras.

It can be observed that the statistical-based methods generally resulted in much poorer performance. The various state-of-the-art learning-based methods, such as the RT, FFCC, and FC4 methods, had much better performance, with the FC4 method having the smallest angular errors. These learning-based methods, however, all have a great number of parameters and require a large memory size, so that they cannot be easily deployed in practice. In contrast, the performance of the proposed PCC method was generally comparable to these state-of-the-art learning-based methods, though it was designed for pure color images. More importantly, the proposed PCC method requires a small memory size, with the number of parameters only around 400. Thus, the proposed PCC method can be applied to both pure color images and normal images, and it can be deployed in practice with a small memory size.

Table 3. Summary the Performance of the Various Methods for the Cube+ Dataset, with the Learning-Based Methods (i.e., Quasi, FFCC, and C5) Trained Using the NUS-8 Dataset

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
WP [2]	9.69	7.48	8.56	1.72	20.49
GW [3]	3.52	2.55	2.82	0.60	7.98
SoG [4]	3.22	2.12	2.44	0.43	7.77
Quasi [26]	2.69	1.76	2.00	0.49	6.45
FFCC [17]	2.69	1.89	2.08	0.46	6.31
C5 ($m = 1$) [15]	2.60	1.86	2.10	0.55	5.89
PCC	2.64	2.08	2.23	0.81	5.41

5. DISCUSSION

A. Rationale of Feature Selection and Network Design

The design of the proposed PCC method included the feature selection and the design of the MLP neural network. The four color features were selected based on the combinations of four statistical-based methods [i.e., white patch, grey world, bright, and PCA (bright and dark)]. In other words, the color features are indeed considered important for illuminant estimation. Different numbers of features and alternative features were also tried, and these four features were found to have the best performance. Regarding the network design, though increasing the number of hidden layers can possibly result in better performance, it will also require greater computational power and memory size. We tried different designs of the network and found the proposed network illustrated in Fig. 4 can achieve

Table 2. Summary of the Performance of Various Methods, in Terms of the Angular Errors, on Two Datasets, CC2018 and NUS-8, and the Number of Parameters for the Learning-Based Methods[a]

	CC2018					NUS-8					
Methods	Mean	Med.	Tri.	Best 25%	Worst 25%	Mean	Med.	Tri.	Best 25%	Worst 25%	No. of Parameters
WP	7.55	5.68	6.35	1.45	16.12	10.62	10.58	10.49	1.86	19.45	-
GW	6.36	6.28	6.28	2.33	10.58	8.42	7.05	7.37	2.41	16.08	-
SoG	4.93	4.01	4.23	1.14	10.20	3.40	2.57	2.73	0.77	7.41	-
Bright	3.98	2.61	-	-	-	3.17	2.41	2.55	0.69	7.02	-
PCA	3.52	2.14	2.47	0.50	8.74	2.92	2.04	2.24	0.62	6.61	-
CM (19 Edge)	3.12	2.38	2.59	0.90	6.46	3.03	2.11	2.25	0.68	7.08	-
RT	2.42	1.65	1.75	0.38	5.87	2.36	1.59	1.74	0.49	5.54	31.5M
CCC	1.95	1.22	1.38	0.35	4.76	2.38	1.48	1.69	0.45	5.85	700
FFCC	1.99	1.31	1.43	0.35	4.75	2.06	1.39	1.53	0.39	4.80	8200
CNN	2.36	1.98	-	-	-	-	-	-	-	-	0.15M
FC4	1.77	1.11	1.29	0.34	4.29	2.12	1.53	1.67	0.48	4.78	4.34M
PCC	2.64	1.61	1.83	0.45	6.62	2.32	1.66	1.80	0.51	5.29	400

[a]The results of FFCC were from [17], and the others were from [8].

the best balance between the performance and complexity. It is worthwhile to further investigate whether advanced neural network strategies (e.g., a shortcut connection [25]) can achieve better results, in terms of performance and complexity, in the future.

B. Cross-Sensor Performance

DNN-based methods are generally trained using images captured by a certain camera or a group of cameras. Since the spectral sensitivity functions of each camera are not identical, the trained models may be device-dependent and can only be applied to the same camera(s). This is a serious challenge in practice, as it requires significant efforts to capture new images and perform model training when a different camera is used.

The proposed PCC method only uses four color features, which can be considered as an upgrade version of statistical-based methods. Thus, these four color features could be considered as device-independent [15]. Therefore, this method was hypothesized to have a good cross-sensor performance. This was validated by training the model using the NUS-8 dataset and testing the model using the Cube+ dataset, using the methods (i.e., C5 [15] and Quasi [26]) that were found to have the good cross-sensor performance and the proposed PCC method, as summarized in Table 3 [2–4,15,17,26]. It can be observed that the performance of the PCC method was comparable to the C5 and Quasi methods, but the number of parameters is around 1/10 of that of the C5 method. Figure 7 illustrates the small variations of the four color features, in terms of the chromaticities, of the PCC method of the same scene captured using three sensors.

Furthermore, such a cross-sensor performance was hypothesized to be more serious in pure color images since the distributions of the image pixel chromaticities would have much larger differences among different cameras in comparison to the normal images. This was validated by training the proposed PCC method using the NUS-8 dataset while testing the performance on other datasets (i.e., Cube+, CC2018, TAU, and PolyU Pure Color), with the performance summarized in Table 4. It can be observed that the PolyU Pure Color dataset resulted in the worst performance, suggesting the uniqueness and also the great value of this dataset for future computational color constancy work.

Table 4. Summary and Comparison of the Performance of the Proposed PCC Method When the Training Was Performed Using the NUS-8 Dataset and the Testing Was Performed Using Each of the Four Datasets

Datasets	Mean	Med.	Tri.	Best 25%	Worst 25%
Cube+	2.64	2.08	2.23	0.81	5.41
CC2018	4.65	3.85	3.91	1.49	9.37
TAU	4.04	3.47	3.62	1.71	7.34
PolyU Pure Color	5.71	4.17	4.62	1.36	12.12

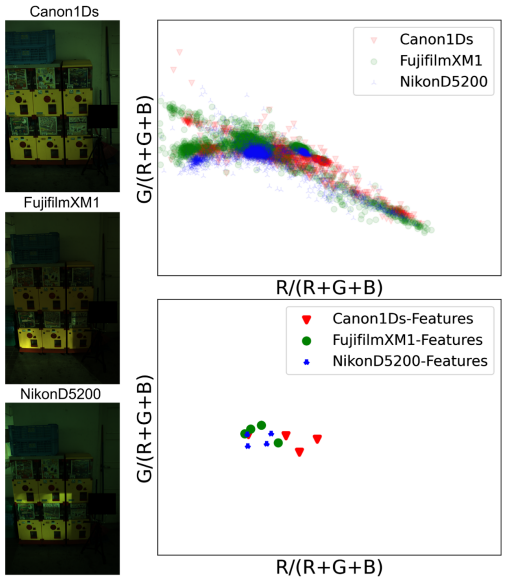

Fig. 7. Chromaticities of the image captured using three different sensors (i.e., Canon 1D, FujifilmXM1, and Nikon D5200) and the four color features, in terms of the chromaticities, derived from each image. This illustrates that the color features have much smaller variations, in comparison to all the pixels in the images, suggesting the good cross-sensor performance of the PCC method (note: the images were selected from the NUS-8 dataset).

6. CONCLUSION

Illuminant estimation is critically important to imaging systems and attracts great attention, but little effort has been made on images that are dominated by a single color (i.e., pure color images). In this study, a pure color image dataset—"PolyU Pure Color" dataset—was developed, and a lightweight feature-based MLP neural network model—PCC—was developed for illuminant estimation. It estimates the chromaticities of the illuminant based on four important color features (i.e., four sets of chromaticities) of an image, including the chromaticities of the maximal, mean, brightest, and darkest pixels in an image.

The proposed PCC method was found to have good performance in estimating the illuminant of pure color images, in terms of the angular error, in comparison to the various statistical-based and learning-based methods. Moreover, its performance on normal images in the existing datasets (i.e., CC2018 and NUS-8) was also comparable to the state-of-the-art learning-based methods, though it was purposely designed for pure color images. Further analyses clearly showed that the proposed PCC method also had a good cross-sensor performance, which is a great challenge for practical deployments. Last but not least, the excellent performance of the proposed PCC method was achieved with a significantly smaller number of parameters (i.e., around 400 parameters) in comparison to the state-of-the-art methods (i.e., FFCC and FC4 methods), taking only 0.25 ms to process an image with an unoptimized Python implementation, which allows it to be deployed in practice.

APPENDIX A

The thumbnails of the images included in the dataset are shown in Fig. 8. The dataset includes outdoor scenes (e.g., green grass, blue sky, and flowers) and indoor scenes (e.g., fabrics and walls).

Funding. Research Grant Council (PolyU 152031/19E).

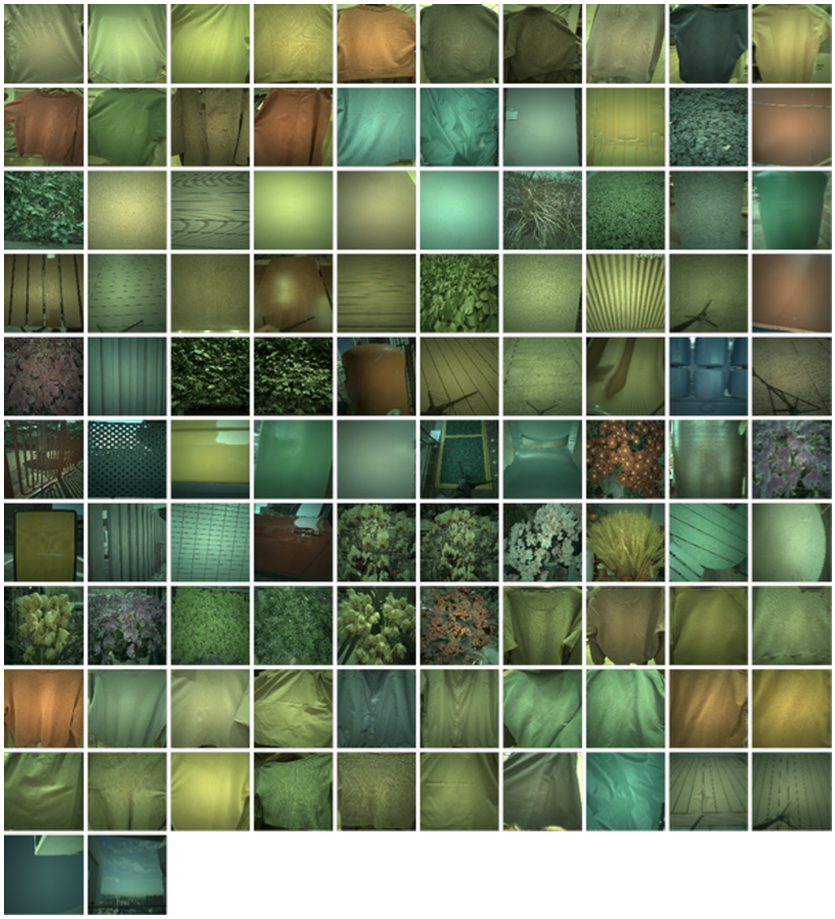

Fig. 8. Thumbnails of the images that were collected and included in PolyU Pure Color dataset.

Disclosures. The authors declare no conflicts of interest.

Data availability. The dataset and code are available in Ref. [27].

REFERENCES

1. A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: survey and experiments," IEEE Trans. Image Process. **20**, 2475–2489 (2011).
2. E. H. Land, "The Retinex theory of color vision," Sci. Am. **237**, 108–129 (1977).
3. G. Buchsbaum, "A spatial processor model for object colour perception," J. Franklin Inst. **310**, 1–26 (1980).
4. G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Color and Imaging Conference* (2004), pp. 37–41.
5. H. R. Vaezi Joze, M. S. Drew, G. D. Finlayson, and P. A. Troncosoey, "The role of bright pixels in illumination estimation," in *Color and Imaging Conference* (2012), pp. 41–46.
6. D. Cheng, D. K. Prasad, and M. S. Brown, "Illuminant estimation for color constancy: why spatial-domain methods work and the role of the or distribution," J. Opt. Soc. Am. A **31**, 1049–1058 (2014).
7. S. Bianco, C. Cusano, and R. Schettini, "Color constancy using CNNs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (IEEE, 2015), pp. 81–89.
8. 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* (IEEE, 2018), pp. 4085–4094.
9. P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2008), pp. 1–8.
10. B. Nikola, K. Koščević, and S. Lončarić, "Unsupervised learning for color constancy," arXiv, arXiv:1712.00436 (2017).
11. F. Laakom, J. Raitoharju, J. Nikkanen, A. Iosifidis, and M. Gabbouj, "Intel-TAU: a color constancy dataset," IEEE Access **9**, 39560–39567 (2021).
12. L. Shi and B. Funt, "Re-processed version of the Gehler color constancy dataset," Computational Vision Lab, School of Computing Science, Simon Fraser University (2022), https://www2.cs.sfu.ca/~colour/data/shi_gehler/.
13. G. Hemrit, G. D. Finlayson, A. Gijsenij, P. V. Gehler, S. Bianco, B. Funt, M. Drew, and L. Shi, "Rehabilitating the colorchecker dataset for illuminant estimation," in *Color and Imaging Conference* (2018), pp. 350–353.
14. Y. C. Lo, C. C. Chang, and H. C. Chiu, "CLCC: contrastive learning for color constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 8053–8063.
15. M. Afifi, J. T. Barron, C. LeGendre, Y. T. Tsai, and F. Bleibel, "Cross-camera convolutional color constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 1981–1990.
16. J. T. Barron, "Convolutional color constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2010), pp. 379–387.
17. J. T. Barron and Y. T. Tsai, "Fast Fourier color constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017), pp. 886–894.
18. H. Gong, "Convolutional mean: a simple convolutional neural network for illuminant estimation," arXiv, arXiv:2001.04911 (2020).

19. G. D. Finlayson, "Corrected-moment illuminant estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2013), pp. 1904–1911.
20. D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Effective learning-based illuminant estimation using simple features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 1000–1008.
21. A. Abdelhamed, A. Punnappurath, and M. S. Brown, "Leveraging the availability of two Cameras for illuminant estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 6637–6646.
22. A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: an imperative style, high-performance deep learning library," arXiv, arXiv:1912.01703 (2019).
23. D. P. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," arXiv, arXiv:1412.6980 (2014).
24. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv, arXiv:1711.05101 (2017).
25. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 770–778.
26. S. Bianco and C. Cusano, "Quasi-unsupervised color constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), pp. 12212–12221.
27. S. Yue and M. Wei, "Color constancy from a pure color view," GitHub (2023), https://github.com/shuwei666/Color-Constancy-PCC.