https://doi.org/10.2352/CIC.2022.30.1.35

Dive into illuminant estimation from a pure color view

Shuwei Yue, Minchen Wei*
Color, Imaging, and Illumination Laboratory, The Hong Kong Polytechnic University
**** minchen.wei@polyu.edu.hk***

Abstract

Illuminant estimation is critically important in computational color constancy, which has attracted great attentions and motivated the development of various statistical- and learning-based methods. Past studies, however, seldom investigated the performance of the methods on pure color images (i.e., an image that is dominated by a single pure color), which are actually very common in daily life. In this paper, we develop a lightweight feature-based Deep Neural Network (DNN)model—Pure Color Constancy (PCC). The model uses four color features (i.e., chromaticity of the maximal, mean, the brightest, and darkest pixels) as the inputs and only contains less than 0.5k parameters. It only takes 0.25ms for processing an image and has good cross-sensor performance. The angular errors on three standard datasets are generally comparable to the state-of-the-art methods. More importantly, the model results in significantly smaller angular errors on the pure color images in PolyU Pure Color dataset, which was recently collected by us.

Introduction

Computational color constancy aims to remove the color cast of illumination in an image, so that the colors appear as they are captured under a canonical illuminant. Therefore, the estimation of the scene illuminant is critically important, which is generally achieved using statistical- or learning-based methods [1]. Estimating the illuminant of a scene is actually an ill-posed problem, since the solution lacks uniqueness and stability. Thus, the various methods and algorithms generally make some assumptions about the statistical characteristics of the expected illuminants and/or the colors of stimuli in a scene. Conventional statistical-based methods typically make assumptions about the colors of stimuli in a scene, such as white patch [2], gray world [3], shades of gray [4], bright [5], and PCA-based [6] shed variant reflectance assumptions. Though these methods can be implemented very efficiently in practice, their assumptions are commonly violated in real scenes, which results inaccurate illuminant estimations and poor image quality. For example, when a captured scene is dominated by one single color, all the above assumptions would be violated.

Recently, Deep Neural Network (DNN) learning-based methods have been applied to the color constancy problem, which has been found to have good performance. Bianco et al [7] was the first to show the great potential to apply a DNN-based model on illuminant estimation. A captured scene was divided into many small regions, and the scene illuminant is estimated based on the estimations made from each small region. Hu et al [8] further found that the illuminants estimated from the ambiguous regions (i.e., a region that is dominated by a single pure color) could be significantly different from those estimated from the valuable regions (i.e., the regions that contain semantic information). This is mainly because a given set of camera *rgb* values could be produced by a surface having such *rgb* values under a white illuminant, or by a white surface under an illuminant with such *rgb* values.

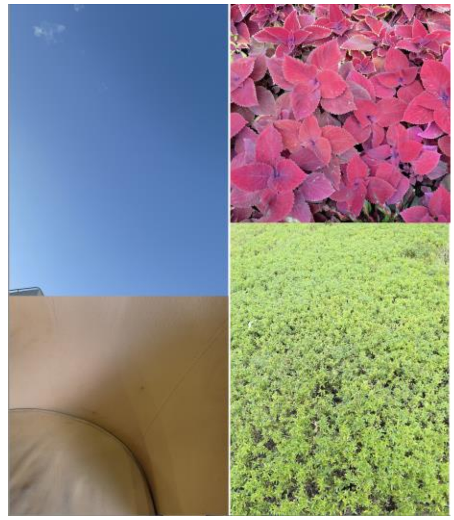

Figure 1. Examples of pure color images (i.e., an image that is dominated by a single pure color), which are contained in "PolyU Pure Color" dataset collected by us. (note: these images are from the cameras directly)

Such ambiguous regions could also happen to the entire captured scene, with the entire image containing a pure color (i.e., pure color images), making the images lack semantic information for estimating the illuminant. Though such pure color images commonly happen in daily life, as shown in Figure 1, and the illuminant estimation is non-trivial for these images, none of the past studies have investigated the performance of the methods on pure color images, and none of the existing models was developed by specifically considering the characteristics of pure color images.

With the above in mind, we propose a *Pure Color Constancy* (PCC) method, a lightweight feature-based DNN model, to significantly improve the performance on pure color images without affecting the performance on typical scenes containing a single illuminant.

The PCC method

Inspired by the "Convolutional Color Constancy" (CCC) method [9], which identifies a 2-D log-chromaticity as a critical feature for illuminant estimation, we aim to estimate the illuminant color based on important color features. Though a similar concept was adopted in a recent work [10], which used a regression tree learning-based method with four features (i.e., the average chromaticities, the brightest color chromaticities, the dominant color chromaticities, and the chromaticity mode of the color palette), suggesting the effectiveness of using simple features for illuminant estimation. This method, however, requires a great number of parameters and computational power, making it impossible for real-world deployment.

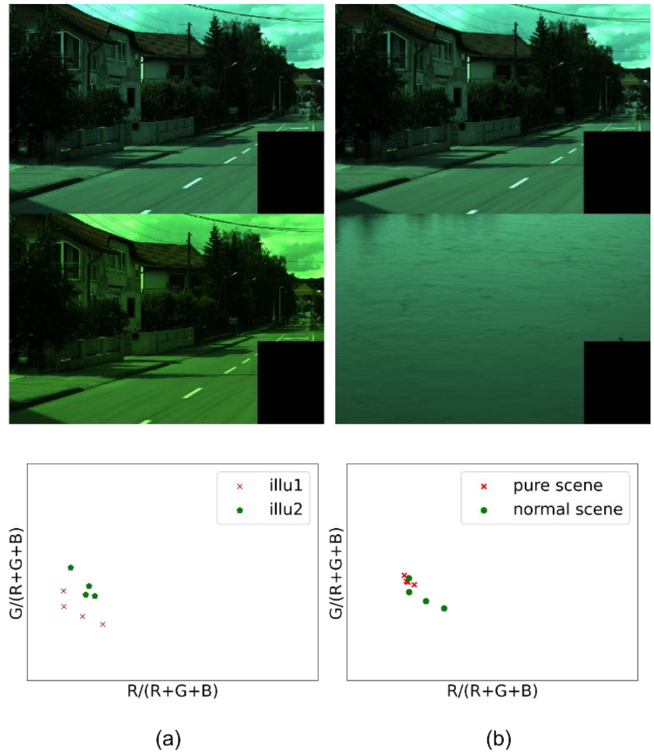

Figure 2. Illustration of the four important color features—normalized brightest, maximal, mean, and darkest chromaticities. (a) An identical scene under two illuminants with large differences of the four features; (b) Comparison between a normal color image and a pure color image, with the latter having the four features clustered together.

Our method was developed based on an observation that the normalized maximal, mean, brightest, and darkest values of each channel vary with the illuminant, as illustrated in Figure 2(a). Therefore, the four features can be used to estimate the illuminants. Moreover, a pure color image tends to have these four features clustered together in the chromaticity space, as illustrated in Figure 2(b). Therefore, the PCC model was designed to have four color features, in terms of normalized chromaticity $\{r, g\} = \{R, G\}/(R + G + B)$, as the inputs, so that they are intensity invariant [10]. These four chromaticity features are: (1) Max chromaticity: the chromaticity of maximum RGB values, $\{r_{max}, g_{max}\} = \{R_{max}, G_{max}\}/(R_{max} + G_{max} + B_{max})$ (2) Mean chromaticity: the chromaticity of mean RGB values, (3) Bright chromaticity: the chromaticity of the pixel having the largest R+G+B values, and (4) Dark chromaticity: the chromaticity of the pixel having the smallest R+G+B values. These four features also correspond to the assumptions of the four widely used illuminant estimation algorithms (i.e., white patch [2], gray world [3], bright [5], and PCA-based methods [6]).

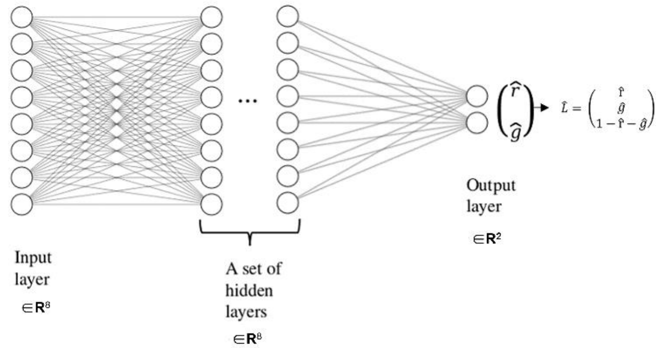

Figure 3. Structure of the proposed PCC model, using the four chromaticity features as the inputs. Each hidden layer has a corresponding ReLU operation.

A lightweight neural network model was then designed, which was inspired by [11]. The model only contains two or five dense layers, with each layer containing only eight neurons, and a corresponding ReLU operation. There are around 150 and 400 parameters for the two- and five-layer networks respectively. The output of the model is the estimated chromaticity (r,g) of the illuminant in the 2D chromaticity color space, with b calculated as 1-r-g. Figure 3 shows the structure of the proposed PCC model.

The proposed PCC based on an unoptimized GPU version only takes 0.25 ms to process an image, which is around 12× faster than the FFCC method [9] and 100× faster than the FC4 method [8]. Moreover, the number of the parameters is around 20 to 50 times smaller than that of the FFCC and around 10,000 times smaller than that of the FC4 method. Error! Reference source not found.**Experiment**

Settings

The PCC network was trained in PyTorch [12], and Adam [13] was adopted as the optimization algorithm, with a learning rate of 10^{-3}. The batch size was set to 1, and the number of epoch was set to 8,000. The standard angular error, as calculated using Eq (1), was adopted as the loss function.

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

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

Datasets and data augmentation

The network was implemented on three datasets, RECommend (REC-2018) Color Checker dataset [14], NUS-8 dataset [10], and Cube+ dataset [17]. The REC-2018 dataset, which is an updated version of Gehler-Shi dataset [15,16], contains 568 indoor and outdoor images captured using two cameras (i.e., Canon 1D and Canon 5D). The NUS-8 dataset contains 1736 images captured by eight different cameras, with each camera capturing around 210 images. The Cube+ dataset contains 1707 images captured by a Canon 550D camera.

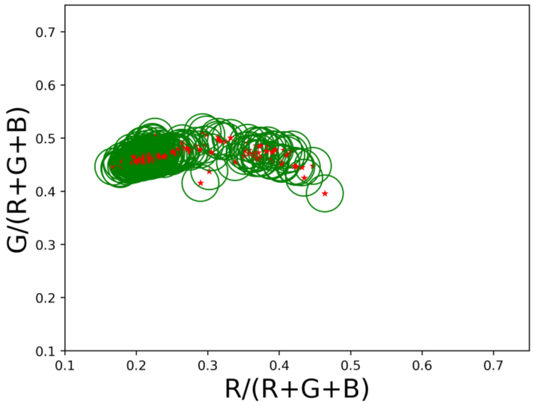

Figure 4. Illustration of how the augmented illuminants were selected. The red dots are the chromaticities of a series of illuminants selected from a real training set. The augmented illuminants were randomly derived by limiting the chromaticity distance between the augmented and selected illuminants to be smaller than 0.01, which is labeled as the green circle.

In order to further increase the number of images, data augmentation was applied using an AWB-aug method, which was based on two past studies [8,18]. The AWB-aug was applied by multiplying a 3 × 3 diagonal matrix $\boldsymbol{M}$ with the diagonal entries of $[\boldsymbol{r_a/r_o, g_a/g_o, b_a/b_o}]$ to an original image $\boldsymbol{I_o}$ to derive an augmented image $\boldsymbol{I_a}$ (i.e., $\boldsymbol{I_a} = \boldsymbol{I_o} \times \boldsymbol{M}$). Since the empirical values used in the past work (e.g., FC4 [8] used random RGB values for data augmentation) may lead to some illuminants that do not exist in reality, we carefully selected a series of illuminants from all the real illuminants captured in the corresponding dataset. Based on these selected illuminants, the augmented illuminants were randomly derived by limiting the chromaticity difference to be smaller than 0.01, as illustrated in Figure 4.

The standard three-folded validation method was used on these three datasets with the data augmentation.

Data analyses and results

The performance of the proposed PCC model was compared with other widely used methods, such as White Patch(WP) [2], Gray World (GW) [3], Shades of Gray (SoG) [4], Bright Pixels with (Bright) [5], Cheng PCA (PCA) [6], Corrected Moment with 19 colors (CM) [17], Regression Tree (RT) [10], CCC [20], FFCC with model Q [9], CLCC [18], based on the standard angular error between the estimated and ground-truth illuminants [1], with the mean, median (Med.), trimean (Tri), best 25%, and worst 25% values reported.

Tables 1, 2, and 3 summarize the performance of the various methods and the proposed PCC model (the results in Tables 2-5 are all based on the 5-layer PCC model) for the three datasets. It should be noted that the training and evaluation were performed on each of the eight cameras in the NUS-8 datasets, while the values reported in Table 2 are the average values across all the eight cameras. For the Cube+ dataset, two additional methods—Color Dog (CD) [21] and Color Beaver (CB) [22]—were also included in the analyses and comparison.

In Table 1, we also summarize the number of the parameters of each method. It can be observed that the size of the proposed PCC model is significantly smaller than that of the state-of-the-art methods, while the performance was relatively similar.

Table 1 Summary and comparison of the model performance, in terms of the angular error, on the REC-2018 Color Checker dataset. (Note: the results of the various methods are extracted from [8] and [18]). The last column summarizes the number of the parameters in each method.

Method	Mean	Med.	Tri.	Best 25%	Worst 25%	No. of Para
WP	7.55	5.68	6.35	1.45	16.12	-
GW	6.36	6.28	6.28	2.33	10.58	-
SoG	4.93	4.01	4.23	1.14	10.20	-
Bright	3.98	2.61	-	-	-	-
PCA	3.52	2.14	2.47	0.50	8.74	-
CM	2.86	2.04	2.22	0.70	6.34	57
RT	2.42	1.65	1.75	0.38	5.87	31.5M
CCC	1.95	1.22	1.38	0.35	4.76	0.7K
FFCC	1.99	1.31	1.43	0.35	4.75	8.2K
Bianco CNN	2.36	1.98	-	-	-	0.15M
FC4	1.77	1.11	1.29	0.34	4.29	4.34M
CLCC [18]	1.44	0.92	1.04	0.27	3.48	1.73M
PCC (2-layer)	**2.92**	**2.07**	**2.32**	**0.68**	**6.93**	**0.15K**
PCC (5-layer)	**2.49**	**1.59**	**1.82**	**0.45**	**5.98**	**0.4K**

Table 2 Summary and comparison of the model performance on the NUS-8 dataset. (Note: the results of the various methods are extracted from [8] and [18]).

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
WP	10.62	10.58	10.49	1.86	19.45
GW	8.42	7.05	7.37	2.41	16.08
SoG	3.40	2.57	2.73	0.77	7.41
Bright	3.17	2.41	2.55	0.69	7.02
PCA	2.92	2.04	2.24	0.62	6.61
CM	3.03	2.11	2.25	0.68	7.08
RT	2.36	1.59	1.74	0.49	5.54
CCC	2.38	1.48	1.69	0.45	5.85
FFCC	2.06	1.39	1.53	0.39	4.80
FC4	2.12	1.53	1.67	0.48	4.78
CLCC [18]	1.84	1.31	1.42	0.41	4.20
PCC	**2.29**	**1.57**	**1.74**	**0.49**	**5.30**

Table 3. Summary and comparison of the model performance on the Cube+ dataset. (Note: the results of the various methods are extracted from [24]).

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
WP	9.69	7.48	8.56	1.72	20.49
GW	7.71	4.29	4.98	1.01	20.19
SoG	2.59	1.73	1.93	0.46	6.19
CD	3.32	1.19	-	0.22	10.22
CB	1.49	0.77	-	0.21	3.94
FFCC	1.38	0.74	0.89	0.19	3.67
C5	1.39	0.79	0.93	0.24	3.55
PCC	**1.79**	**1.02**	**1.23**	**0.25**	**4.61**

Generalization

Since the spectral sensitivity functions generally vary with cameras, the traditional CNN-based models are device-dependent.

The proposed PCC model, however, only uses four simple color features as the inputs. As the variations of these four features with the cameras are relatively smaller, the model should be able to be used across cameras.

This was validated by training the model using the images captured by a NIKON-D40 camera in the NUS dataset, with the performance evaluated on the images in the Cube+ dataset without any processing. Table 4 compares the performance of the various methods, including the state-of-art C5 [23] and Quasi [24] methods. It is worthwhile to point out that the number of the parameters in the PCC model is only around 1/10 of that in the C5 model, but the results are generally comparable.

Table 4 Summary and comparison of the model performance for generalization on the Cube+ dataset. The three smallest values are highlighted with a gray background.

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
WP	9.69	7.48	8.56	1.72	20.49
GW	3.52	2.55	2.82	0.60	7.98
SoG	3.22	2.12	2.44	0.43	7.77
Quasi	2.69	1.76	2.00	0.49	6.45
FFCC	2.69	1.89	2.08	0.46	6.31
C5(m=1)	2.60	1.86	2.10	0.55	5.89
C5(m=7)	2.10	1.38	1.40	0.41	4.97
PCC	**2.19**	**1.23**	**1.53**	**0.41**	**5.50**

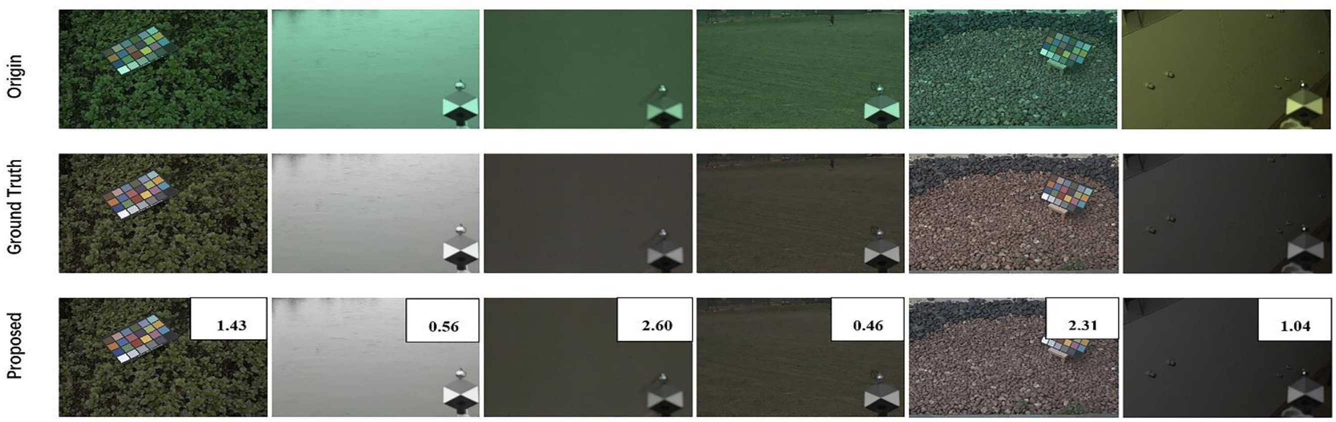

Figure 5. Examples of pure color images in the NUS-8 and Cube+ datasets. The values are the angular errors derived using the proposed PCC model.

Performance on "PolyU Pure Color" dataset

The datasets mentioned above have a small number of pure color images. Figure 5 shows some examples from the NUS-8 and Cube+ datasets, with the angular errors derived using the proposed PCC model being labeled, which are much smaller than those derived using the other conventional methods.

We built a dataset "PolyU Pure Color", which contains pure color images captured using a Huawei P50 Pro smartphone. Currently, the dataset contains 102 pure color images, including outdoor scenes (e.g., green grass, blue sky, flowers) and indoor scenes (e.g., fabrics and walls under illumination). The images shown in Figure 1 are from this dataset.

Table 5 Summary and comparison of the model performance on the PolyU Pure Color dataset, a newly collected dataset containing pure color images.

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 (AlexNet)	3.83	3.02	3.47	1.16	8.03
PCC	**2.91**	**1.39**	**1.83**	**0.42**	**7.30**

The training and evaluation were repeated on this dataset, with Table 5 summarizing the performance of the proposed PCC model, together with other methods. It can be observed that the performance of the PCC model is significantly better than the other methods, including the FC4 [8] method.

Conclusion

We propose a PCC model for illuminant estimation. The model is a feature-based lightweight DNN model, which uses four color features as the inputs, including the chromaticity of maximal, mean, the brightest, and darkest pixels in an image. In particular, the model was leveraged from a learning-based method in the log-chroma domain [20], and an important observation that the maximal and mean values of the RGB channels in a pure color image should be very similar. Thus, it was expected to have much better performance for pure color images, which was verified using a recently collected dataset (i.e., PolyU Pure Color dataset), and also for other typical images, which was verified using the three existing datasets (i.e., REC-2018 Color Checker, NUS-8, and Cube+ datasets). Also, the model has a good cross-sensor performance. More importantly, such a good performance is achieved with around 0.2~0.4k parameters in the model, and only takes 0.25 ms for processing an image with an unoptimized Python implementation.

Reference

[1] A. Gijsenij, T. Gevers, and J. Van De Weijer, “Computational color constancy: survey and experiments,” IEEE Trans. Image Process., 20, 9, (2011).

[2] E. H. Land, “The retinex theory of color vision,” Sci. Am., 237, 6, (1977).

[3] G. Buchsbaum, “A spatial processor model for object colour perception,” J. Franklin Inst., 310, 1, (1980).

[4] G. D. Finlayson and E. Trezzi, Shades of gray and colour constancy, Color and Imaging Conference, pg. 37 (2004).

[5] H. R. Vaezi Joze, M. S. Drew, G. D. Finlayson, and P. A. Troncoso ey, The role of bright pixels in illumination estimation, Color Imaging Conf., pg. 41, (2012).

[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, 5, (2014).

[7] S. Bianco, C. Cusano, and R. Schettini, Color constancy using CNNs, Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pg. 81 (2015).

[8] Y. Hu, B. Wang, and Stephen. Lin, FC4: Fully convolutional color constancy with confidence-weighted pooling, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 4085 (2017).

[9] J. T. Barron and Y. T. Tsai, Fast fourier color constancy, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 6950 (2017).

[10] D. Cheng, B. Price, S. Cohen, and M. S. Brown, Effective learning-based illuminant estimation using simple features, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 1000 (2015).

[11] A. Abdelhamed, A. Punnappurath, and M. S. Brown, Leveraging the availability of two Cameras for illuminant estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 6637 (2021).

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N.Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: an imperative style, high-performance deep learning library,” Advances in neural information processing systems., 32 (2019).

[13] D. P. Kingma and B. Jimmy, “Adam: a method for stochastic optimization,” arXiv preprint arXiv. (2014).

[14] 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, Color Imaging Conf, pg. 350 (2018).

[15] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, Bayesian color constancy revisited, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 1 (2008)

[16] L. Shi and B. Funt, Re-processed version of the gehler color constancy dataset. https://www2.cs. sfu.ca/~colour/data/shi_gehler/, (2019)

[17] B. Nikola, K. Koščević, and S. Lončarić, “Unsupervised learning for color constancy,” arXiv preprint arXiv. (2017).

[18] Y. C. Lo, C. C. Chang, and H. C. Chiu, CLCC: Contrastive learning for color constancy, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pg. 8053 (2021).

[19] G. D. Finlayson, Corrected-moment illuminant estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 1904 (2013).

[20] J. T. Barron, Convolutional color constancy, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 629 (2010).

[21] B. Nikola and S. Loncaric， Color dog-guiding the global Illumination estimation to better accuracy， VISAPP， pg. 129 (2015).

[22] K. Koščevic, N. Banic, and S. Loncaric, Color beaver: Bounding illumination estimations for higher accuracy, Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl, pg. 183 (2019).

[23] M. Afifi, J. T. Barron, C. LeGendre, Y. T. Tsai, and F. Bleibel, Cross-camera convolutional color constancy, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 1981 (2021).

[24] S. Bianco and C. Cusano, Quasi-unsupervised color constancy, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pg. 12212 (2019).

Author Biography

Shuwei Yue is a PhD student at Color, Imaging, and Illumination Laboratory at The Hong Kong Polytechnic University. He received his master degree in Digital Medial Technology from Hangzhou Dianzi University in 2018, and worked in DJI as an image algorithm engineer for two years. His research interests are color vision and computational photography.

Minchen Wei is an associate professor, and the director of Color, Imaging, and Illumination Laboratory at The Hong Kong Polytechnic University. His research interests are color science, imaging science, and illumination technologies. He is currently an associate editor of the Journal of the Optical Society of America A, and Color Research & Application.