Color Constancy: How to Deal with Camera Bias?

Yi-Tun Lin[1,2†]
ethanx188@gmail.com
Bianjiang Yang[3]
yang2208@purdue.edu
Hao Xie[1]
horacezju@gmail.com
Wenbin Wang[1]
wenbinw@meta.com
Honghong Peng[1]
hhpeng@meta.com
Jun Hu[1,4‡]
junhu@apple.com

[1] Meta Platforms, Inc.
CA, USA
[2] Colour and Imaging Lab
University of East Anglia
Norwich, UK
[3] Purdue University
IN, USA
[4] Apple Inc.
CA, USA

Abstract

In color constancy, we seek to estimate and remove the color of the illuminating light from the captured raw RGB images. While the learning-based methods perform better than the traditional statistics-based methods, they are typically tuned to perform well on a particular camera, i.e., their performance on images captured by other cameras is subject to their cross-camera generalizability. This problem was partially addressed, in various prior works, using deep learning architectures. In this paper, we examine how these cross-camera models perform compared to where we "pre-calibrate" the camera biases in the training and/or testing images using a simple homographic color correction procedure which can be easily done on the camera manufacturer's part. And further, with pre-calibrated data we examine by how much we could simplify the original cross-camera models. We show that cross-camera color constancy with a simple pre-calibration process yields up to 36% performance boost, which in turn indicates that the original cross-camera methods have only limited ability to compensate for the cross-camera prediction bias. Surprisingly, with this newly proposed evaluation protocol, we also found that some single-camera color constancy algorithms already possess cross-camera ability similar to adopting a camera bias pre-calibration.

1 Introduction

Object colors in the scene can vary drastically under different illumination, and thus it is an important task to remove the effect of lighting from the captured RGB images for

[†] The work was done during Yi-Tun's internship with Meta.
[‡] The work was performed when Jun Hu worked at Meta Platforms, Inc.

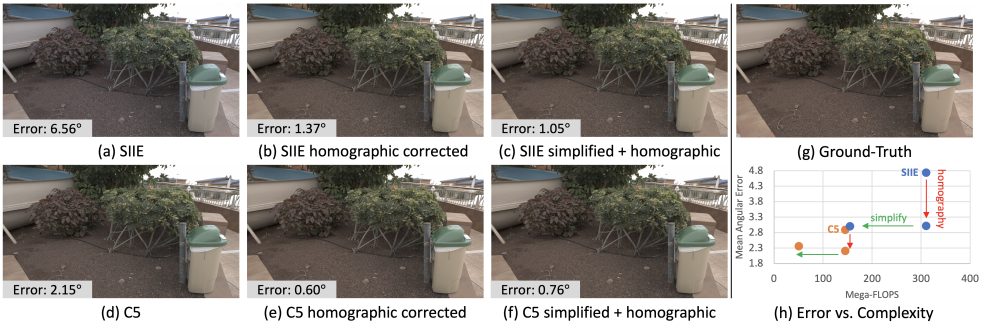

Figure 1: Visualization of the efficacy of homographic pre-correction for cross-camera color constancy. **(a)(d)** The cross-camera predictions by SIIE [1] and C5 [2]. **(b)(e)** SIIE and C5 with homographic pre-correction. **(c)(f)** Simplified SIIE and C5 with homographic pre-correction. **(g)** The ground-truth white balance. **(h)** The models' performance (across 3 cross-camera scenarios) versus complexity.

illumination-invariant computer vision results. In practice, this *color constancy* process includes the first step of estimating the dominant illumination color in a captured raw RGB image—a.k.a. the "white point"—and secondly dividing the image channel-by-channel by the white point color [18]. This second step, called *white balancing*, is usually a fixed process in the camera's image processing pipeline, and therefore in most computer vision studies, the problem of color constancy focuses on estimating the illumination white point from the captured raw RGB image.

Historically, most methods seek to find some statistics of the image's pixel and/or edge colors that could indicate the illumination white point [5, 8, 10, 12, 17, 31, 35, 37]. Advantageously, these statistics-based assumptions usually do not specify the use of any particular camera and thus the performance—i.e., whether the assumptions may succeed or fail—does not depend on the camera. Recently, most of the proposed methods are based on machine learning algorithms, e.g., [6, 7, 14, 21, 27, 32], where raw RGB images with labeled ground-truth white points are used for training. These methods are shown to generally deliver better color constancy compared to the statistics-based methods. However, they are tuned to perform well separately on individual cameras. Several works suggest that under a *cross-camera* evaluation framework (i.e., the images used for training and testing are captured by different cameras), some learning algorithms do not work as well as in the single-camera case [1, 2, 20, 25, 26].

In this paper, we aim to re-evaluate cross-camera color constancy and seek simplifications to the problem in the *manufacturer*'s point of view. Insofar most works on cross-camera color constancy assume that we do not have any prior knowledge about the testing cameras (and therefore a more complicated deep-learning structure is "needed" to compensate for the camera bias). We believe this assumption does not represent most of the real-life use cases where camera devices usually store some characterization data in their hardware memories. We propose that—instead of increasing the model complexities to *partially* solve cross-camera color constancy—manufacturers can consider storing some extra characterization data on cameras that helps resolve this problem. Of course, this data should be light weighted, supporting fast processing, and easily acquired.

We take the inspiration from the related *color correction* research [19], which is a pro-

cess usually conducted after the white balancing step [9]. In color correction, "post-white-balancing" image(s) of the 24-patch Macbeth Color Checker (MCC) [23] under one or more reference illumination(s) are captured by the camera of concern, and (usually) a 3×3 linear-transformation matrix is trained to optimally convert these MCC colors to the standard colors, such as CIE XYZ or linear sRGB [34]. This trained matrix is then stored in the camera's memory and used to convert colors almost universally for any real-world scenes captured. Analogously, for cross-camera color constancy, we could use the "pre-white-balancing" MCC colors to train another color correction matrix. Here we could, given the raw MCC colors of both training and testing cameras, train a color correction matrix that relates the raw colors of the two cameras. Alternatively, a fixed *reference camera* could be assigned (just like the standard color spaces) where both training and testing cameras can map to. As such, the trained color constancy algorithms can continue to work on the images coming from a new testing camera as if they were captured by the same camera.

Figure 1 exhibits the efficacy of adopting a pre-color-constancy color-correction process (we use the color homographic correction [15]). Here, two cross-camera color constancy algorithms, SIIE [1] and C5 [2], are trained on images captured by Canon 5DSR and Sony IMX135, and tested on images of Nikon D810. We see that the original white-point predictions by SIIE and C5 still introduce considerable errors. Then, we show that the errors are mitigated by a homographic correction. Further, with homographic correction, we can simplify SIIE and C5 while retaining the improved performance. Examining the error versus complexity chart in Figure 1(h), we see that the homography process improves the SIIE performance by 36% (24% for C5), and the improved performance retains even when we lower half of its FLOPS [33] complexity (two-third lower for C5).

Converse to our finding that the cross-camera methods are not actually able to fully compensate for the camera bias, we are also surprised to find that, using the same evaluation methodology, some learning-based "single-camera" color constancy algorithms can already compensate for the cross-camera bias almost as good as adopting a homographic pre-correction.

2 Related Works

Color Constancy. Statistics-based color constancy algorithms are based on explainable assumptions, either in terms of light-reflective physics (e.g., white patch [8], gray index [31]) and/or the empirical understanding of scene contents (e.g., gray world [10], black-and-white PCA [12]). These methods do not require learning, however, their assumptions may occasionally *fail* for particular types of scenes.

Generally speaking, learning-based color constancy algorithms perform better than methods based on image statistics. Shallow-learned algorithms [6, 7, 11, 13, 16] use traditional optimization approaches such as least-squares regression or maximum likelihood estimation, with the primary focus being to find effective features that represent individual images, among which FFCC [7] delivers leading performance in many single-camera benchmarks.

It is generally accepted that end-to-end deep-learning architectures deliver state-of-the-art performances. FC4 [21] is a fully-connected architecture adopting a pretrained Alexnet or Squeeze-net. It is the building block of many successor deep-learning methods, including C4 [38], MDLCC [36], and CLCC [27]. CLCC, which proposed to use multiple FC4's in training to contrast the same-scene-different-illuminants and different-scenes-same-illuminant cases in data augmentation, is widely considered the current state-of-the-art

single-camera color constancy method.

Cross-Camera Color Constancy. Recently, several approaches are proposed that aim to mitigate the cross-camera performance degradation of learned color constancy. SIIE [1] is one of the first algorithms that explicitly addressed cross-camera color constancy. In SIIE, a sensor-mapping network is used to predict a 3×3 color correction matrix *from each input image* (there is no guarantee that same-camera images will provide the same matrix). Using this matrix, the original image is transformed into a reference color space before being fed into the illumination-estimation network to deliver the final white point predictions.

Then, C5 [2] is considered the current state-of-the-art for cross-camera color constancy. C5 uses multiple input images (9 images as suggested in the original paper) from the same camera with shared encoders and a cross-encoder max-pooling strategy to learn an FFCC algorithm [7] with corrected histogram input and FFCC model parameters.

There are other approaches that do not use deep learning to solve the cross-camera problem. Koskinen et al. [25] proposes to first obtain the camera-independent high-dimensional spectral representation from the color images captured by the training camera and then back-project the spectra to the RGB space using the target camera sensors' spectral sensitivities. Also assuming the camera sensors' spectral property is known, Gao et al. [20] trains a 3×3 RGB-to-RGB correction mapping using the simulated RGB scenes of both cameras from hyperspectral images. And yet, it is difficult to characterize the cameras' spectral sensitivities. As part of the motivation of our proposed method, we are to show that making use of the 24-patch MCC chart (instead of hyperspectral images and camera spectral sensitivities) is already sufficient to obtain a useful cross-camera mapping.

Color Correction. Most color correction algorithms are trained to map the *post-white-balancing* colors of the MCC chart captured under the same illumination(s) by two different devices (or between one device and a standard color space [34]). Since MCC contains only 24 color patches, i.e., only 24 data points in a typical color correction training, the complexity of the learning algorithm used could not be too high. Traditionally, linear least-squares (LLS) regression is used. To introduce non-linearity in the mapping, polynomial and root-polynomial least-squares (PLS, RPLS) are proposed [19, 29].

More recently, the theory of color homography was proposed by Finlayson et al. [15], where they claimed that object chromaticities—i.e., the R/G and B/G values—under different illuminations and/or captured by different cameras are related by a homographic transform. In effect, the main advantage of using a homography formulation is that, unlike LLS, PLS and RPLS, it removes the effect of shading differences when mapping the color of the same object viewed under different illumination and/or camera conditions [15].

The comparisons between different color correction algorithms used particularly for our cross-camera color constancy experiments are shown in the supplementary materials (Appendix A). In this paper we will use the better-performing color homography method.

3 Method

In this section, we will first establish how the homographic color correction algorithm [15] can be applied to cross-camera color constancy. Then, we will introduce our new evaluation protocol based on the INTEL-TAU dataset [26].

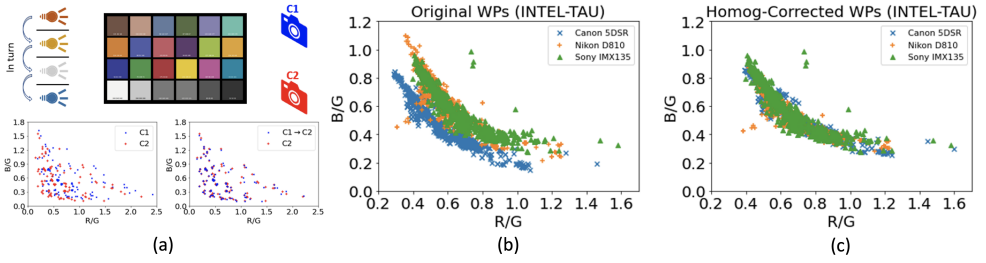

Figure 2: **(a)** Top: MCC chart under different reference lights captured by two cameras. Bottom left: the original chromaticity distributions of two cameras' MCC colors. Bottom right: transformed MCC colors using a trained homography. **(b)** Ground-truth white point distributions of the three cameras in INTEL-TAU [26]. **(c)** The homographic-corrected white point distributions.

3.1 Homographic Color Correction

Let us denote C1 and C2 as two different camera models. We wish to find a fixed color mapping $f_{\text{C1}\rightarrow\text{C2}}$ such that:

$$\begin{bmatrix} R_2/G_2 \\ B_2/G_2 \end{bmatrix} \approx f_{\text{C1}\rightarrow\text{C2}}\left(\begin{bmatrix} R_1/G_1 \\ B_1/G_1 \end{bmatrix} \right), \tag{1}$$

where $[R_1, G_1, B_1]$ and $[R_2, G_2, B_2]$ refer to the camera responses at the same pixel captured by C1 and C2, respectively. According to the color homography theory [15], this mapping is formulated as a fixed homographic transform:

$$\alpha \begin{bmatrix} R_2/G_2 \\ B_2/G_2 \\ 1 \end{bmatrix} \approx \mathbf{H} \begin{bmatrix} R_1/G_1 \\ B_1/G_1 \\ 1 \end{bmatrix}, \tag{2}$$

where $\mathbf{H}$ is a 3×3 learnable matrix. α is a scalar depending on the input R_1/G_1 and B_1/G_1 and is indicated directly by the third element of the derived vector (right-hand-side of the equation). The predicted C2 chromaticities are then the first two elements of the derived vector divided by α.

Of course, the best fitting $\mathbf{H}$ needs to be learned from some *training data*. For this, we propose to use a common 24-patch Macbeth Color Checker (MCC) [28]. As illustrated in Figure 2(a), we use the two concerned cameras, C1 and C2, to capture images of MCC under one or more reference light(s). The colors of the patches are then extracted from the images. We know that these two sets of MCC colors will have some degree of mismatch, as shown in the bottom-left panel of Figure 2(a).

Then, we are to train a homographic mapping that best matches the MCC colors of the two cameras. From Equation (2), we formulate a least-squares minimization:

$$\min_{\mathbf{H}} \sum_{\ell=1}^{L} \sum_{i=1}^{24} || \frac{1}{\alpha^{(i,\ell)}} \mathbf{H} \underline{\mathbf{c}}_1^{(i,\ell)} - \underline{\mathbf{c}}_2^{(i,\ell)} ||_2^2, \tag{3}$$

where $\underline{\mathbf{c}}_1^{(i,\ell)}$ and $\underline{\mathbf{c}}_2^{(i,\ell)}$ are the $[R/G, B/G, 1]^\top$ vectors of the ith MCC patch under the ℓth reference light captured by C1 and C2, respectively. And, $||\cdot||_2$ denotes the $L2$-norm, and

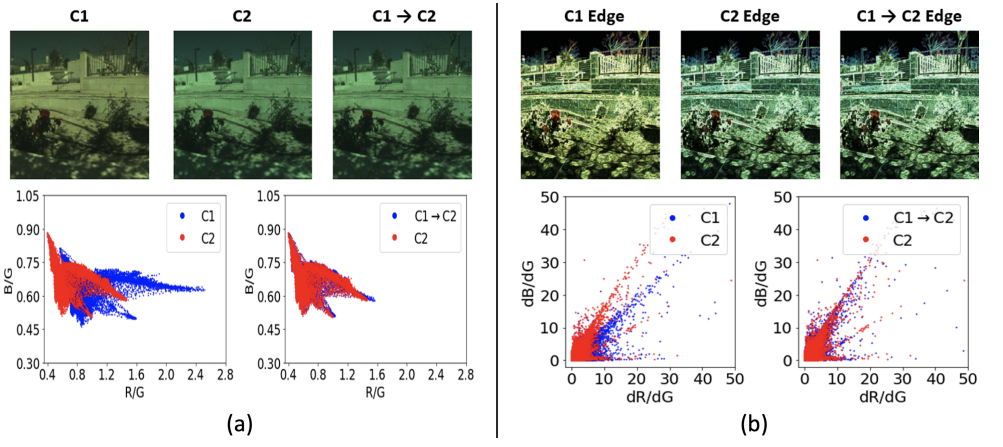

Figure 3: **(a)** Top row: images of the same scene captured by two different cameras, C1 and C2, and the predicted C2 image from C1 colors using a homographic correction (C1→C2). Bottom row: chromaticity distributions of the two original images (left) and distributions of the original and predicted C2 images (right). **(b)** The original and predicted edge images, and the corresponding distribution comparisons, with the Same-*G*-Channel assumption applied.

$\alpha^{(i,\ell)}$ is the third element of $\mathbf{H}\underline{\mathbf{c}}_1^{(i,\ell)}$. The efficacy of the trained mapping operating on the training MCC colors is shown in the bottom-right panel of Figure 2(a).

In practice, Equation (3) can be solved very quickly. For instance, INTEL-TAU [26] provides the MCC colors under 10 reference illuminations, i.e., there are 240 data points for solving this minimization. We use the Powell method [30] implemented in Python's scipy.optimize.minimize function for the minimization, and all training instances (obtaining the optimal homographic mapping between each pair of two cameras) were done in less than 0.2 seconds individually. Note that it is not necessary to use multiple reference illuminations for training: we include an extensive study that uses each illumination in INTEL-TAU individually to train the homographic mapping, which suggests somewhat consistent results for the simplified C5 model. This study is provided in supplementary materials (Appendix B).

As a primary evidence of the effectiveness of the correction, in Figure 2(b), we examine the R/G-B/G chromaticity distribution of the ground-truth white points of the INTEL-TAU database [26]. Here, each camera is indicated by a different color. Clearly, the original white point distributions of individual cameras follow a similar trend (most likely along the Planckian locus [24] of each camera), though subject to varying degrees of shifting in their relative positions. Then, using a 3×3 homographic matrix transform (Equation (2)), we can align all ground-truth white points to roughly the same locus when projected to one selected reference camera, as shown in Figure 2(c).

Same-*G*-Channel Assumption. While many traditional color constancy algorithms can work with only the R/G-B/G chromaticity information, leading methods such as FFCC, C5, and other deep learning algorithms often require the *edge color* information, i.e., the stacked channel-wise spatial derivative images [2, 7]. A homographic correction process shown in Equation (2) only predicts the R and B chromaticity *ratios* with respect to the G-channel values. For edge images we will need to calculate the gradients in the G-channel as well. For

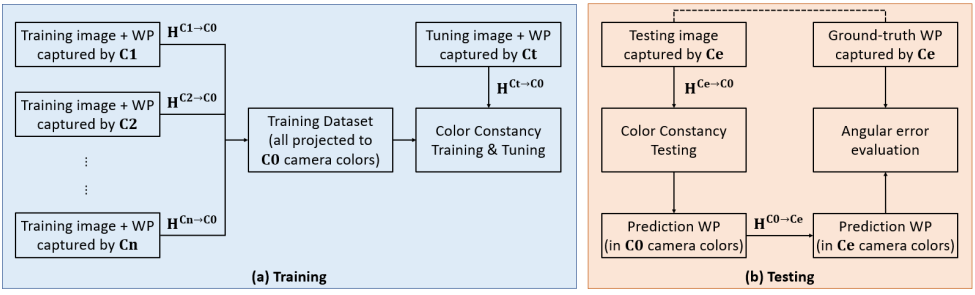

Figure 4: The training and testing schemes of the homographic-corrected cross-camera color constancy. "WP" is short for the image white point. The capital letter "C" followed by a number or a letter indicates individual camera models. The **H** matrices represent homographic color conversions (between the two indicated cameras in their respective superscripts; mapping from the left one to the right one). Both the images' pixel colors and the white points can be converted using **H**, following Equation (2).

this, we choose to simply assume the same G-channel image of C1 (the original camera) for C2 (the target camera), and then the homographic-corrected R/G and B/G ratios for C2 are applied to this borrowed G-channel to derive R- and B-channel images.

Empirically, it is shown in [22] that the G-sensors' spectral sensitivities of their tested 28 cameras are indeed more similar than the R- and B-sensors' sensitivities, with 98.7% of the variance described by the first PCA basis. We also examine the edge-color chromaticities of multiple real-world scenes and cameras, showing that our "same-G-channel" assumption works quite well in predicting the correct edge-color distributions (one example is given in Figure 3, while more results are given in Appendix C of the supplementary materials[1]).

Homographic-Corrected Color Constancy. The training and testing schemes for applying homographic pre-correction for color constancy are shown in Figure 4. Broadly speaking, cross-camera color constancy is considered when the involved images—either used in training, tuning (validation), or testing—are captured by different cameras. The idea is to obtain homographic correction matrices for all possible pairings of involved cameras following the methodology introduced in the last section.

Then, while deciding on one *reference camera* (in the case of Figure 4, the camera "C0"), we are to convert all other images (and their ground-truth white points) to the reference camera's colors using their respective camera-specific homographic corrections, before feeding those images to the color constancy algorithm for training, tuning, and testing. Of course, in the case of testing, we need to convert the predicted white points back to the original testing camera's color space for a standard angular error evaluation.

The decision of which camera to be used as the reference camera can occasionally simplify the process presented in Figure 4. For example, if we see that the majority of the images in training are captured by one particular camera, we can choose this camera to be the reference camera so as to reduce the number of homographic corrections we need to run in training. Alternatively, one could try to use all different cameras in turn as the reference camera, and choose the one that provides the best overall color constancy outcomes.

[1]The displayed raw images are simulated using a scene in the ICVL hyperspectral image dataset [3] and two selected cameras' spectral sensitivity functions in [22].

3.2 INTEL-TAU Dataset and Proposed Evaluation Protocol

There exist many variants of evaluation protocols for cross-camera color constancy, e.g., [1, 2, 25]. In these works, multiple publicly available datasets are gathered, and while leaving one dataset out in turn for testing, the considered algorithms are trained on other datasets altogether. This *cross-dataset* approach is often framed as a cross-camera evaluation.

Indeed, different datasets tend to use different cameras to collect image data. And yet, considering that there are up to 10 different cameras used in the training set [1], different datasets might use different white targets to obtain ground-truth white points (e.g., Cube+ [4] versus NUS [12]), and all other possible cross-dataset variations, it is arguable whether a cross-dataset evaluation protocol indeed evaluates the algorithms' cross-camera applicability, or they might actually focus more on examining their ability to "learn from heterogeneous data". Furthermore, in several datasets there are same-scene images captured by different cameras, including the NUS [12] and INTEL-TAU [26] datasets. Non-careful training/testing splits can cause problematic evaluation results. This said, we still include some testing results using these protocols in Appendix D of our supplementary materials.

Our proposed evaluation protocol uses only the INTEL-TAU dataset [26]. INTEL-TAU uses 3 different cameras: Canon 5DSR, Nikon D810, and Sony IMX135. Each camera is used to collect a unique set of images—different for different cameras—in the "field_1_ cameras" dataset folder (Canon: 1645 images, Nikon: 2329 images, Sony: 1656 images). Then, for each camera there are three other image sets: "field_3_cameras", "lab_printouts" and "lab_realscene". These last three folders include the same 464 scenes for all 3 cameras.

Therefore, to ensure (i) we do not include any testing scenes in training, (ii) to setup a cross-camera evaluation scenario, and (iii) to enforce a 3-fold cross-validation process in training, we propose the following evaluation protocol on INTEL-TAU:

1. Use only the field_1_cameras images of Nikon and Sony as the training set, and conduct 3-fold cross-validation to obtain 3 models.
2. Test these 3-fold models on all images (from all 4 image folders) of Canon.
3. Repeat Step 1 and 2 twice more, while changing the training cameras to Canon and Sony and test on Nikon, and then train on Canon and Nikon and test on Sony.

That is, we conduct 3 (train/test camera splits) $\times$ 3 (cross-validation folds) $=$ 9 experiments per algorithm.

This protocol mimics a practical use case for camera manufacturers where the training dataset is collected using 2 prior camera models, and while a new camera is manufactured, we wish not to retrain the color constancy algorithm but to use them directly on the new camera. Further, from scene 11 to 20 in the lab_realscene dataset folders, we can extract MCC colors under 10 different illumination lights captured by the 3 cameras. This information is used to train the homographic color correction mapping detailed in Section 3.1.

4 Results and Discussion

The evaluation results are shown in Table 1. Like in the prior art [2, 7, 27], we calculate 5 angular error statistics including Mean, Median (Med.), Trimean (Tri.), Average of the best 25% (B.25) and Average of the worst 25% (W.25). The presented numbers are the 5 statistics

Method	Mean	Med	Tri	B.25	W.25	FLOPS
Statistics-based methods						
White Patch [8]	10.33	11.07	10.35	1.53	18.61	
Shades-of-Gray [5]	5.73	4.24	4.75	1.00	12.78	
Gray-World [11]	4.89	3.88	4.15	0.97	10.52	
Black-and-White PCA [3]	4.51	3.16	3.50	0.73	10.60	
Bright Pixels [2]	4.41	3.13	3.45	0.73	10.25	
2nd-order Gray Edge [13]	4.19	3.40	3.57	1.05	8.70	
1st-order Gray Edge [13]	4.16	3.25	3.46	0.94	8.93	
Gray Index [3]	4.00	2.38	2.81	0.52	10.10	
Single-camera shallow-learned methods						
FFCC [9]	3.42	2.75	2.84	0.74	7.37	
FFCC-homog	2.52	1.64	1.83	0.48	6.08	
Single-camera deep-learning methods						
FC4 [21]	2.19	1.41	1.55	0.45	5.31	7.3G
FC4-homog	2.07	1.37	1.52	0.44	4.93	
CLCC [7]	1.97	1.40	1.51	0.45	**1.49**	6.5G
CLCC-homog	**1.94**	**1.30**	**1.42**	**0.41**	4.59	
Cross-camera methods						
SIIE [9]	4.74	4.13	4.25	1.96	8.58	310.0M
SIIE-homog	3.02	1.95	2.23	0.55	7.24	
SIIE-simp-homog	3.01	1.97	2.24	0.55	7.23	154.7M
C5 [9]	2.89	2.27	2.38	0.76	6.16	145.2M
C5-homog	2.21	1.40	1.58	0.42	5.40	
C5-simp-homog	2.36	1.50	1.69	0.44	5.76	50.6M

Table 1: The cross-camera 3-fold cross-validation results on the INTEL-TAU dataset. The performance of the homographic-corrected and the homographic-corrected-simplified algorithms are marked in blue and pink, respectively. For deep-learning-based algorithms, we provide their FLOPS in the last column. The best results are shown in bold and underlined.

individually averaged over the 9 experiments specified in Section 3.2[2]. Visualized results are shown in Figure 5.

Canon 5DSR is set as the reference camera (the "C0" in Figure 4) for all homographic-corrected algorithms[3]. Also, in our simplification attempt for SIIE, we replace the sensor-mapping network with the homographic color correction and only keep the illumination-estimation network for the final predictions. Then, as we use homographic corrections to correct the cross-camera input instead, we simplify C5 by only using 1 image at a time as input—hence we need to use only 1, not 9, input encoder.

First, under the cross-camera scenario, we see that learning-based algorithms in general still work better than the statistics-based methods, except for the original SIIE. We reckon this is because the sensor-mapping network in SIIE does not restrict that images captured by the same camera should infer the same or similar color correction matrix, which leads to severe overfitting to the training cameras.

Next, we see that all considered learning-based algorithms are improved by our homographic pre-corrections, with FFCC, C5 and SIIE improving more significantly than FC4 and CLCC. As the latter two methods are much more complex than the former three, it is likely that FC4 and CLCC are more able to support ambitious data augmentation (that drastically perturbs the ground-truth white points) which aids their cross-camera performances [14]. Their limitation, however, is that if a new camera has very different spectral sensitivities [22], the data augmentation may fail to apply (where homographic correction might bring in even better margins—to be confirmed in the future work).

Finally, it is evident that with the help of homographic corrections, we can greatly simplify C5 and SIIE while retaining similarly good results. Compared to using extra networks to learn cross-camera color constancy, we believe our proposed approach is much simpler and better performing, while still being easily accessible (it is entirely feasible to re-use the same MCC data captured for post-white-balancing color correction here).

[2]Per-camera error statistics are supplied in the supplementary materials (Appendix E).

[3]We also tested the simplified C5 using Nikon D810 and Sony IMX135 as reference cameras (Appendix F).

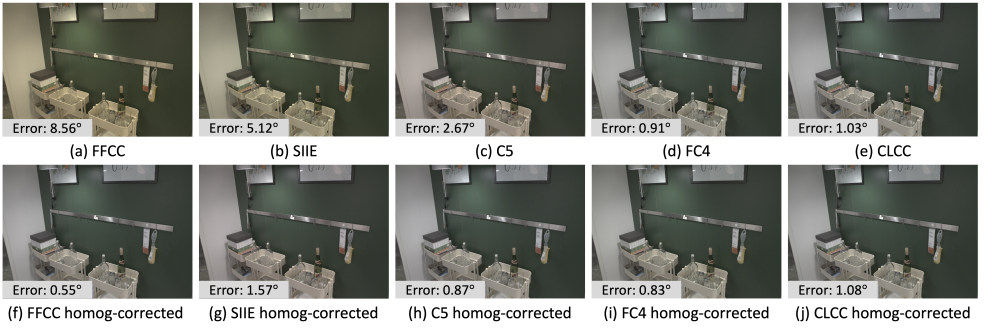

Figure 5: **(a)-(e)** Visualization of results of the original color constancy algorithms. Each visualized image is white-balanced using the corresponding predicted white point, and then applied with a gamma correction. **(f)-(j)** Results of the homographic-corrected methods.

5 Conclusion

We propose to apply a simple pre-white-balancing homographic color correction step which can usefully compensate for the camera device biases before training and/or testing the color constancy algorithms. We show that this simple added process (very fast to train and to run) can improve state-of-the-art cross-camera color constancy performance, and can replace a significant part of deep-network structures of the leading cross-camera models [1, 2]. Our results also show that some leading single-camera color constancy algorithms [21, 27] are more capable in a cross-camera scenario than the cross-camera methods in the prior art.

References

[1] Mahmoud Afifi and Michael S Brown. Sensor-independent illumination estimation for dnn models. In *British Machine Vision Conference*, 2019.

[2] Mahmoud Afifi, Jonathan T Barron, Chloe LeGendre, Yun-Ta Tsai, and Francois Bleibel. Cross-camera convolutional color constancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1981–1990, 2021.

[3] Boaz Arad and Ohad Ben-Shahar. Sparse recovery of hyperspectral signal from natural rgb images. In *European Conference on Computer Vision*, pages 19–34. Springer, 2016.

[4] Nikola Banić, Karlo Koščević, and Sven Lončarić. Unsupervised learning for color constancy. *arXiv preprint arXiv:1712.00436*, 2017.

[5] Kobus Barnard, Vlad Cardei, and Brian Funt. A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data. *IEEE transactions on Image Processing*, 11(9):972–984, 2002.

[6] Jonathan T Barron. Convolutional color constancy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 379–387, 2015.

[7] Jonathan T Barron and Yun-Ta Tsai. Fast fourier color constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–894, 2017.

[8] David H Brainard and Brian A Wandell. Analysis of the retinex theory of color vision. *Journal of the Optical Society of America A*, 3(10):1651–1661, 1986.

[9] Michael S Brown and SJ Kim. Understanding the in-camera image processing pipeline for computer vision. In *IEEE International Conference on Computer Vision – Tutorial*, volume 3, 2019.

[10] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1–26, 1980.

[11] Ayan Chakrabarti. Color constancy by learning to predict chromaticity from luminance. *Advances in Neural Information Processing Systems*, 28, 2015.

[12] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *Journal of the Optical Society of America A*, 31(5):1049–1058, 2014.

[13] Dongliang Cheng, Brian Price, Scott Cohen, and Michael S Brown. Effective learning-based illuminant estimation using simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1008, 2015.

[14] Ilija Domislović, Donik Vršnak, Marko Subašić, and Sven Lončarić. One-net: Convolutional color constancy simplified. *Pattern Recognition Letters*, 159:31–37, 2022.

[15] Graham Finlayson, Han Gong, and Robert B Fisher. Color homography: theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1): 20–33, 2017.

[16] Graham D Finlayson. Corrected-moment illuminant estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1904–1911, 2013.

[17] Graham D Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, volume 2004, pages 37–41. Society for Imaging Science and Technology, 2004.

[18] Graham D Finlayson, Mark S Drew, and Brian V Funt. Diagonal transforms suffice for color constancy. In *International Conference on Computer Vision*, pages 164–171. IEEE, 1993.

[19] Graham D Finlayson, Michal Mackiewicz, and Anya Hurlbert. Color correction using root-polynomial regression. *IEEE Transactions on Image Processing*, 24(5):1460–1470, 2015.

[20] Shao-Bing Gao, Ming Zhang, Chao-Yi Li, and Yong-Jie Li. Improving color constancy by discounting the variation of camera spectral sensitivity. *Journal of the Optical Society of America A*, 34(8):1448–1462, 2017.

[21] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4085–4094, 2017.

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

[23] Hamid Reza Vaezi Joze, Mark S Drew, Graham D Finlayson, and Perla Aurora Troncoso Rey. The role of bright pixels in illumination estimation. In *Color and Imaging Conference*, volume 2012, pages 41–46. Citeseer, 2012.

[24] Young-sun Kim, Bong-Hwan Cho, Bong-Soon Kang, and Doo-Il Hong. Color temperature conversion system and method using the same, April 4 2006. US Patent 7,024,034.

[25] Samu Koskinen, Dan Yang, and Joni-Kristian Kämäräinen. Cross-dataset color constancy revisited using sensor-to-sensor transfer. In *British Machine Vision Conference*, 2020.

[26] Firas Laakom, Jenni Raitoharju, Jarno Nikkanen, Alexandros Iosifidis, and Moncef Gabbouj. Intel-tau: A color constancy dataset. *IEEE Access*, 9:39560–39567, 2021.

[27] Yi-Chen Lo, Chia-Che Chang, Hsuan-Chao Chiu, Yu-Hao Huang, Chia-Ping Chen, Yu-Lin Chang, and Kevin Jou. Clcc: Contrastive learning for color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8053–8063, 2021.

[28] Calvin S McCamy, Harold Marcus, James G Davidson, et al. A color-rendition chart. *Journal of Applied Photographic Engineering*, 2(3):95–99, 1976.

[29] Rang Nguyen, Dilip K Prasad, and Michael S Brown. Raw-to-raw: Mapping between image sensor color responses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3398–3405, 2014.

[30] Michael JD Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964.

[31] Yanlin Qian, Joni-Kristian Kamarainen, Jarno Nikkanen, and Jiri Matas. On finding gray pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8062–8070, 2019.

[32] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *European Conference on Computer Vision*, pages 371–387. Springer, 2016.

[33] Vladislav Sovrasov. ptflops: a flops counting tool for neural networks in pytorch framework. https://github.com/sovrasov/flops-counter.pytorch. 2018-2023.

[34] Sabine Süsstrunk, Robert Buckley, and Steve Swen. Standard rgb color spaces. In *Color and Imaging Conference*, volume 1999, pages 127–134. Society for Imaging Science and Technology, 1999.

[35] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214, 2007.

[36] Jin Xiao, Shuhang Gu, and Lei Zhang. Multi-domain learning for accurate and few-shot color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3258–3267, 2020.

[37] Kai-Fu Yang, Shao-Bing Gao, and Yong-Jie Li. Efficient illuminant estimation for color constancy using grey pixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2254–2263, 2015.

[38] Huanglin Yu, Ke Chen, Kaiqi Wang, Yanlin Qian, Zhaoxiang Zhang, and Kui Jia. Cascading convolutional color constancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12725–12732, 2020.