To be published in Journal of the Optical Society of America A:

Title: Effective cross-sensor color constancy using a dual-mapping strategy

Authors: Shuwei Yue,Minchen Wei

Accepted: 15 December 23

Posted 15 December 23

DOI: https://doi.org/10.1364/JOSAA.505814

Effective cross-sensor color constancy using a dual-mapping strategy

Shuwei Yue[1] **and Minchen Wei**[1,2*]

[1] *Color, Imaging, and Illumination Laboratory, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
[2] *Color, Imaging, and Metaverse Research Center, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
[*] *minchen.wei@polyu.edu.hk*

Abstract:
Deep Neural Networks (DNNs) have been widely used for illuminant estimation, which commonly requires great efforts to collect sensor-specific data. In this article, we propose a dual-mapping strategy—DMCC method. It only requires the white points captured by the training and testing sensors under a D65 condition to reconstruct the image and illuminant data, and then maps the reconstructed image into sparse features. These features, together with the reconstructed illuminants, were used to train a lightweight multi-layer perceptron (MLP) model, which can be directly used to estimate the illuminant for the testing sensor. The proposed model was found to have performance comparable to other state-of-the-art methods, based on the three available datasets. Moreover, the smaller number of parameters, faster speed, and not requiring data collection using the testing sensor make it ready for practical deployment. This article is an extension of [1], with more detailed results, analyses, and discussions.

1. Introduction

Color constancy is the ability of the human visual system to maintain the color appearance of objects relatively constant under different illuminants [2]. Computational color constancy algorithms are developed to emulate such an ability for digital cameras, with the estimation of illuminant being the key challenge. This is also referred to as auto white balance (AWB), which is an important step in the camera image signal processor (ISP) pipeline.

The various algorithms can be classified into statistical- and learning-based methods. Statistical-based methods, such as the gray-world method [3], typically estimate the illuminants based on individual images. They are generally simple and do not rely on the spectral sensitivities of the cameras, but the performance is also limited. In comparison, learning-based methods, such as gamut mapping [4] and color moment-based [5] methods, have better performance. In recent years, Deep Neural Network (DNN)-based methods [6–8] were developed. They can lead to better performance than the various learning-based methods.

DNN-based methods generally treat illuminant estimation as a regression task, deriving the illuminant based on the input image data:

$$\mathbf{L}_i = f^{\theta}(\mathbf{Y}_i) \tag{1}$$

where $\mathbf{Y}_i$ is the linear RAW-RGB image dataset, $\mathbf{L}_i$ is the corresponding illuminants dataset, i and θ are the image sample index and learning parameters respectively. The function $f(\cdot)$ is the model, learning the relationship between the images and illuminants in a training dataset.

DNN-based models always need to be trained for each individual camera sensor, since the relationship between the images and illuminants significantly varies with the spectral sensitivity functions of sensors, as illustrated in Fig. 1(A). With the data of the training sensor labeled as the source domain $D_s = \{\mathbf{L}_{s,i}, \mathbf{Y}_{s,i}\}$ and the data of the testing sensor as the target domain

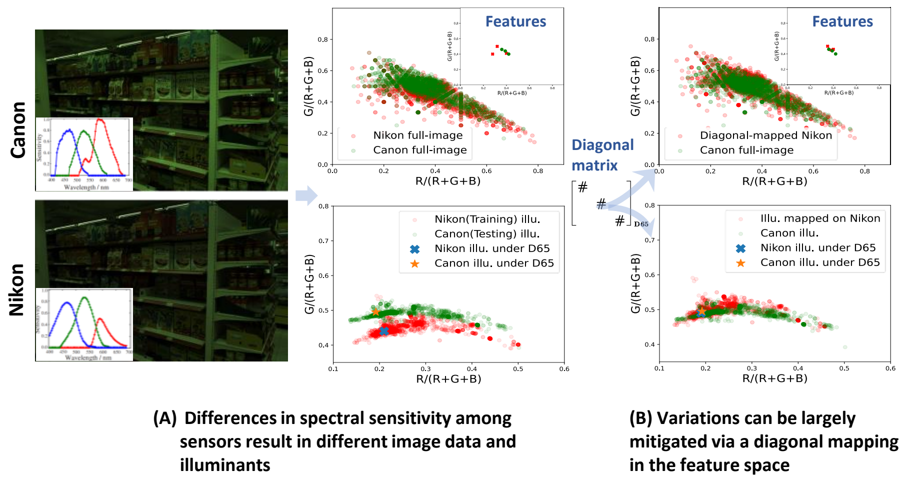

Fig. 1. Illustration of the proposed dual-mapping strategy. (A) An example of a scene captured by two different camera sensors−Nikon and Canon, exhibits variations in image data and illuminant distributions. (B) Illustration of the effectiveness of using a diagonal mapping strategy in a feature space to reduce the variations.

$D_t = \{\mathbf{L}_{t,i}, \mathbf{Y}_{t,i}\}$, Eq. 1 can be expressed as follows:

$$\begin{aligned} \mathbf{L}_{s,i} &= f^{\theta_s}(\mathbf{Y}_{s,i}) \\ \mathbf{L}_{t,i} &= f^{\theta_t}(\mathbf{Y}_{t,i}) \end{aligned} \tag{2}$$

where $f^{\theta_s} \neq f^{\theta_t}$ due to $D_s \neq D_t$. Therefore, a model that is trained using one sensor cannot be directly applied to another sensor if the spectral sensitivity functions of the two sensors are different. Such a problem is not non-trivial, as it requires great efforts to collect both image and illuminant data and to train a new model when a new sensor is used, which becomes a serious challenge for deploying DNN-based methods for practical applications.

1.1. Prior Works

Efforts have been made to address the cross-sensor problem of DNN-based methods, which can be classified into two categories based on their practical applications: (i) model re-training-free (MRTF) methods and (ii) data re-collection-free (DRCF) methods.

Model Re-Training-Free (MRTF) methods These methods aim to develop a universal model that can be directly applied to other sensors without re-training [9–11], or with just a little fine-tuning [11–13]. This is of interest to both academia and industry since it minimizes the efforts of data collection and re-training. The underlying concept is that the training is performed on a wide range of datasets in various domains, such as RAW-RGB images captured by different sensors or even distinct color spaces [11], with multi-task learning [12]. This can be expressed as $D_t \subset D_s$, suggesting that a well-trained model f^{θ_s} has a great potential to have a good performance on a testing set D_t.

The universal models, however, are difficult to train due to the difficulty in mastering multi-domain datasets, which commonly results in highly complicated models and thus impossible to be deployed in practice. More importantly, overfitting commonly happens to these models, especially when large variations exist in datasets due to the wide range of sensors and other factors (e.g., lenses).

One of the most recent state-of-the-art methods (i.e., C5 [10]) leverages hypernetworks [14] and the principles of convolutional color constancy (CCC) [15], and considers the fast fourier color constancy (FFCC) [16] for ensuring the reliable performance on various sensors. By incorporating hypernetworks, the C5 method dynamically adjusts the weightings according to the variations of the input content, ensuring adaptability to various imaging conditions. The good performance of the C5 method is introduced by the diverse and large training datasets, which include labeled and unlabeled images captured by multiple sensors. To fine-tune the model for a certain testing sensor, only a few images captured by the testing sensor are needed and no label information is needed. The optimal number of images for achieving the best performance, however, varies from sensor to sensor. This introduces another hyperparameter, making the method more complicated and difficult for practical deployment. Moreover, complicated data preprocessing steps, such as the log histogram operation in terms of spatial and gradient aspects, also make it difficult for practical deployment.

To increase the size of the training data, Bianco and Cusano [11] innovatively include sRGB images from the internet in the training data, with the model being directly deployed (or with fine-tuning) on the RAW-RGB testing data. This is based on the assumption that the sRGB images can be considered white-balanced, with a 'quasi-unsupervised' strategy used to train a DNN model to detect achromatic pixels based on the grayscale images. Such a method not only effectively increases the size of the training data, but also allows the application to images captured by any sensor. The heavy network and the unsatisfactory performance, however, are the main weaknesses.

Different from the previous 'learning-aware' methods, a 'color-aware' method called SIIE was proposed by Afifi et al [9]. It learns an 'XYZ-like' color space in an end-to-end manner to construct the MRTF model. The assumption of the existence of an independent working space derived through a simple transformation matrix for all cameras, however, may not be valid. This can be observed from the diminished results derived based on the data from a sensor that was significantly different from the training sensor. Similar to the methods discussed above, this method also leads to overfitting.

In addition to the methods that are completely re-training-free, methods that adopt few-shot fine-tuning strategies are also available. We classify these methods into the MRTF category as well, since they also aim to create a universal model. The only difference is that minor adjustments are made for a specific testing sensor based on a small number of images, which does not require too much effort for data re-collection. McDonagh et al. [13] was the first to apply a meta-learning few-shot strategy (i.e., MAML [17]) on the cross-sensor color constancy problems. The method establishes initial model parameters during the meta-learning phase for optimizing the performance on unseen tasks. It makes it vital to define tasks that cover a wide range of scenarios. Specifically, the tasks are defined based on an assumption that images with similar white point color temperatures would have similar dominant colors. Tuning the hyperparameters of the MAML model, however, is challenging and time-consuming due to its complexity. Inspired by this idea and the FC4 [6] framework, Xiao et al. [12] propose a multi-task learning method (i.e., MDLCC), which includes two modules—the common feature module and the sensor-specific reweight module. Though the shared feature extractor model can effectively learn from the images captured by different sensors and thus increase the size of the training data, the method requires a high memory and becomes difficult for practical deployment.

To sum up, though MRTF methods generally provide promising solutions to cross-sensor color constancy, they still have weaknesses (e.g., overfitting and complexity) for deployment. Therefore, researchers are looking for possibilities to focus on individual testing sensors instead of all sensors together, and the methods are considered data recollection-free (DRCF).

Data Re-Collection-Free (DRCF) methods These methods can be considered as special types of MRTF methods. Instead of aiming to train a universal model that works for all sensors, these methods aim to train a model for a specific sensor, which can significantly reduce the effort of data re-collection.

Such an approach directly trains a model f^{θ_t} for the testing data, primarily using the source data D_s, which achieves better model performance on the testing data, lower likelihood of overfitting, and a relatively lightweight model design. This, however, is achieved at the expense of an obvious drawback that a distinct model needs to be trained for each individual testing sensor.

Currently, there are only a few DRCF methods. One method was developed based on the Bayesian [18] framework and was designed to have the ability to handle multi-task images. It uses the illuminants captured by the testing sensors as the ground truth, trains the RAW images captured by different sensors as the input data, and employs a Bayesian-based CNN framework, which leads to good performance. The necessity to collect the testing illuminants, however, becomes a challenge. On one hand, these illuminants are needed for constructing the training labels. On the other hand, a comprehensive estimation of the illuminants is critical for tuning the hyperparameters of the clustering algorithms, which adds complexity to the process.

In this article, we propose an illumination estimation method—Dual Mapping Color Constancy (DMCC)—for cross-sensor applications, which addresses the challenges of the existing methods. It does not require great efforts in data re-collection, but can also result in performance that is comparable to the state-of-the-art methods. This is achieved using a dual mapping strategy, including a diagonal mapping and a feature mapping (extraction), as illustrated in Fig. 1(B). Moreover, this proposed method is easy to train, quick to implement, and memory-efficient, making it a practical solution to be deployed on ISP chips.

2. Proposed Method

2.1. Problem Formulation

The proposed DMCC method has two main phases, as illustrated in Fig. 2. In the calibration phase, a diagonal matrix M is derived based on the two white points, with one captured by the training and testing camera sensors respectively, under a D65 condition. In the training phase, images and illuminants from the training sensor are reconstructed using the matrix M, which allows to directly train f^{θ_t} using the data pairs in the training domain $\{\mathbf{Y}_s, \mathbf{L}_s\}$, without requiring data recollection using the testing sensor. Following this, a feature extractor is used to map the reconstructed full image data $M \times \mathbf{Y}_s$ into sparse features. This process, as illustrated in Fig. 1 (B), was found to make the mapped features from the training and testing data well aligned:

$$g(M \times \mathbf{Y}_s) \sim g(\mathbf{Y}_t) \tag{3}$$

where $g(\cdot)$ is the feature extractor. It was also found that the distribution of the reconstructed illuminants, derived using the calibration matrix M and the illuminants captured by the training sensor $\mathbf{L}_s$, was well aligned with that of the testing sensor, as shown in Fig. 1 (B):

$$M \times \mathbf{L}_s \sim \mathbf{L}_t \tag{4}$$

Then, a multi-layer perceptron (MLP) model f^{θ_t} can be trained using $\{g(M \times \mathbf{Y}_s), M \times \mathbf{L}_s\}$, which can be expressed as follows (note: the multiplication sign $\times$ is omitted for simplicity):

$$\theta_t^* = \arg\min_{\theta_t} \sum_{i=1}^{n} L(M\mathbf{L}_{s,i}, f^{\theta_t}(g(M\mathbf{Y}_{s,i}))) \tag{5}$$

where i is the image index, n is the total number of training images, and $L(\cdot)$ is the loss function.

Although it is impossible to have a perfect alignment between each individual pair of training and testing data, our proposed method is able to effectively reduce the discrepancy. The efficacy of employing $\{g(\mathbf{Y}, \mathbf{L})\}$ to train f^{θ} has been revealed in our recent work [7].

Feature extractor $g(\cdot)$ In this context, $g(\cdot)$ represents an abstract function encapsulating the process of extracting a set of features (i.e., the maximum, mean, brightest, and darkest pixels) in terms of the chromaticities $\{r, g\} = \{R, G\}/(R + G + B)$ from the image data $\mathbf{Y}$. The effectiveness of using $\{r, g\}$ chromaticities has been validated in past studies (e.g., [5,7,19]), due to its effectiveness in mitigating illumination variations and reducing dimensionality. The specific expressions for these features are as follows: $\{R_{\text{max}}, G_{\text{max}}\} \Rightarrow \{r_{\text{max}}, g_{\text{max}}\}$, $\{R_{\text{mean}}, G_{\text{mean}}\} \Rightarrow \{r_{\text{mean}}, g_{\text{mean}}\}$, $\{R_b^p, G_b^p\} \Rightarrow \{r_b, g_b\}$ (where $p = \text{argmax}(R_i + G_i + B_i)$, and $\{R_d^p, G_d^p\} \Rightarrow \{r_d, g_d\}$ (where $p = \text{argmin}(R_i + G_i + B_i)$. Through such a feature extraction operation, the image data is mapped into four sparse features, which are then used as the inputs for the MLP.

In summary, the proposed DMCC method combines the mapping of data using a calibration matrix M and the mapping of image data using a feature extractor $g(\cdot)$, which effectively reduces the domain discrepancy caused by the sensors.

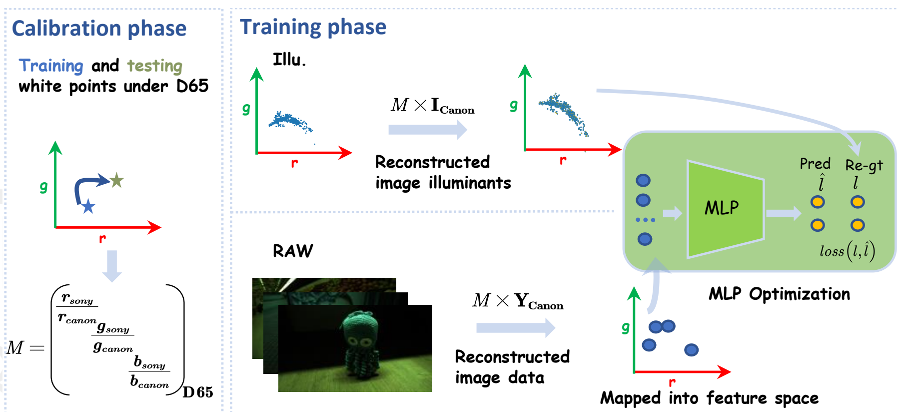

Fig. 2. Overview of the proposed DMCC method. The calibration phase derives the diagonal matrix M using the two white points captured by the training (i.e., Canon) and testing (i.e., Sony) sensors under a D65 condition. The diagonal matrix M is used to reconstruct the image data in the training data (i.e., $M \times \mathbf{Y}_{\text{Canon}}$) and the training illuminants (i.e., $M \times \mathbf{I}_{\text{Canon}}$). Features are then extracted from the reconstructed image data, which is used to train an MLP model with the reconstructed illuminants.

2.2. Architecture of DMCC

The DMCC method includes a diagonal mapping and a feature mapping followed by an MLP training. In particular, the feature mapping is similar to the PCC method proposed in our recent study [7], with modifications made on the hyperparameters, particularly the number of neurons per layer was increased from 8 to 11 and the total number of parameters became around 800. The output of the model is the estimated illuminant chromaticities $(\hat{r}, \hat{g})$ in the 2-D chromaticity color space, with $\hat{b}$ calculated as $1 - \hat{r} - \hat{g}$. Such an MLP-based network has a fast inference time. With an unoptimized Python implementation, it only takes ~0.3 and ~1.0 ms to process an image on an RTX3070Ti GPU and Intel-i9 CPU respectively, which is ~25 times faster than the fastest existing cross-sensor color constancy method (i.e., the C5 method) and has ~700 times fewer parameters. Also, training the DMCC model from scratch only takes less than an hour with the hardware specifications described above. These make the method suitable for practical deployments.

3. Experiment

3.1. Implementation Details

Loss function The proposed DMCC method adopts the traditional angular error between the estimated illuminant $\hat{\boldsymbol{\ell}}$ and the ground truth illuminant $\boldsymbol{\ell}$ with a regularization as the loss function:

$$\boldsymbol{L}(\theta) = cos^{-1}\left(\frac{\boldsymbol{\ell} \odot \hat{\boldsymbol{\ell}}}{||\boldsymbol{\ell}|| \times \left\|\hat{\boldsymbol{\ell}}\right\|}\right) + \lambda||\theta||_1 \tag{6}$$

where $\odot$ represents the inner products and $cos^{-1}(\cdot)$ is the inverse of a cosine function. L1 regularization is employed to adjust the training parameters θ to avoid overfitting, and λ is the regularization weight parameter whose number is 10^{-5}.

Settings The DMCC framework, constructed with PyTorch and integrated with CUDA support, uses the Adam optimizer [20] for training, in conjunction with He initialization [21]. We utilize a batch size of 32 over $10,000$ epochs with a learning rate of 7×10^{-3}. In addition, a cosine annealing strategy [22], is applied to adjust the learning rate, and an early stopping strategy is employed to save the best-performing model throughout the training process. All the hyperparameters used are kept the same in the experiment. A grid search strategy was used to identify the optimal parameters using the data captured by one camera (i.e., a Sony camera) coupled with a Canon white point under D65 conditions, both of which are included in the INTEL-TAU dataset [23], as described in Section 4. The optimization process utilized a standard three-fold cross-validation method, which was considered simple and effective, and also used in past work (e.g., [24] and [7]). It is worthwhile to investigate whether advanced methods, such as nested cross-validation, can further improve the results.

White points The diagonal matrix M is derived from the white points captured by the training and testing sensors under a D65 condition. The publicly available datasets, however, only include the white point of the RAW images in the camera color space, which does not allow to derive the correlated color temperature (CCT) of the illuminant and the diagonal matrix M. Thus, we used the two pre-calibrated matrices, C_1 and C_2, and a trial-and-error strategy, as described in the supplementary material, to estimate the CCT of each image.

3.2. Data Augmentation and Preprocessing

In order to further improve the accuracy of reconstructing the training set using the diagonal matrix M, AWB-Aug [7] was employed to perform the data augmentation, which involved an illuminant enhancement strategy. Specifically, a uniform sampling around the illuminant was performed in the chromaticity space, with the illuminant positioned at the center of the circle. The radius of the circle, a hyperparameter, was set to 0.05, which was found to produce stable results, as shown in Fig. 3. The data augmentation method used here seems to be effective for training the DMCC model, but other methods can also be tried in future studies.

In the experiment, linear RAW-RGB images, with the calibration labels and black level subtracted, were used. Also, saturated and dark pixels were clipped. Moreover, since the method is based on sparse features and is resolution-independent, the images were resized to $64 \times 64 \times 3$ and normalized for fast processing.

3.3. Datasets

In our experiments, all the cameras in the three different datasets (i.e., INTEL-TAU [23] (three cameras), NUS-8 [25] (eight cameras), and Cube+ datasets [26] (one camera)) were used.

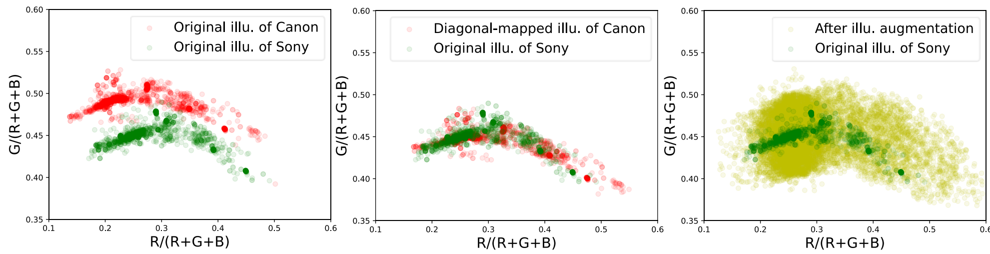

Fig. 3. Illustration of the effectiveness of the data augmentation to cover the variations of the illuminants in the testing dataset. Left: the original distribution of the illuminants in the training and testing sets; Middle: the changes introduced by the diagonal matrix mapping; Right: the improved similarity of the distributions of the illuminants between the training and testing sets after the data augmentation.

A modified three-fold validation approach, with two folds for training and the other fold for validation, was used. In particular, the training and validation processes were conducted solely on the training sensor (and a white point from the testing sensor). This was performed to lead to the best performance on the testing sensor, instead of pursuing an optimal performance for all the sensors. Five statistical results, including the mean, median (Med.), trimean (Tri.), the mean of the smallest 25% (Best 25%), and the mean of the largest 25% (Worst 25%) of the angular error between the estimated and the ground-truth illuminants, in terms of degrees, were used to characterize the performance.

4. Results and Discussions

4.1. Comparative Results

INTEL-TAU dataset (Table 1) The INTEL-TAU evaluation strategy was used to allow a fair comparison. In other words, for testing the data of the Sony sensor, the training and validation were performed using the data of the Canon sensor. Similarly, the data of the Nikon sensor was used for training and validation, when that of the Canon sensor was used for testing; that of the Sony sensor was used for training and validation when that of the Nikon sensor was used for testing. It can be observed that the proposed DMCC method outperformed all the statistical-based algorithms and most DNN-based methods. In particular, its performance was generally comparable to the state-of-the-art C5 method and was roughly on par when the C5 method uses a single image data without a label (m=1). The DMCC method, however, uses an image label without the image data, and it only requires 1/700 execution time and 1/25 memory usage in comparison to the C5 method.

Cube+ and NUS-8 datasets (Table 2) When evaluating the Cube+ and NUS-8 datasets, the model was trained solely on the INTEL-TAU Sony IMX135 dataset and tested on the Cube+ and NUS-8 datasets. In particular, the training sensor (i.e., Sony IMX135) was carefully selected to have the illuminants ***far away*** from the testing sensors' illuminants. Table 2 shows that our DMCC method outperformed the C5(m=1) methods. Moreover, the training set only included the INTEL-TAU Sony IMX135 dataset, without similar image data or illuminant-related training (please refer to the supplementary material for details), which highlights the superior robustness and generalizability of the DMCC method. Figure 4 shows some examples of the images processed by the various methods. In this article, we used a single training dataset and different testing datasets, with the performance suggesting the strong adaptability of the proposed DMCC method. Further studies using different training datasets with a same testing dataset will be also

Table 1. Summary of the performance of various methods, in terms of angular errors, on the INTEL-TAU datasets, together with the processing time and parameter size. The results of the Gray-World, White Patch, Shades-of-Gray, and Cheng-PCA were extracted from [23], and those of the Quasi-Unsupervised, SIIE, FFCC, C5, and MDLCC were extracted from [10] and [12]. The proposed method is highlighted in yellow.

INTEL-TAU Dataset	Best25%	Mean	Med.	Tri.	Worst25%	Time(ms)/Size(MB)
Gray-world [3]	0.9	4.7	3.7	4.0	10.0	- /-
White-Patch [2]	1.1	7.0	5.4	6.2	14.6	- / -
Shades-of-Gray [27]	0.7	4.0	2.9	3.2	9.0	- /-
Cheng-PCA [25]	0.7	4.6	3.4	3.7	10.3	- /-
Quasi-Unsupervised CC [11]	0.7	3.7	2.7	2.9	8.6	90 / 622
SIIE [9]	0.7	3.4	2.4	2.6	7.8	35 / 10.3
MDLCC [12]	-	-	-	-	-	25 / 6
C5(m=7) [10]	0.5	2.6	1.7	-	6.2	7 / 2.09
C5(m=1) [10]	0.7	3.0	2.2	-	6.7	7 / 2.09
DMCC (Ours)	0.7	3.0	2.3	2.2	6.8	0.3 / 0.003

interesting. For the NUS-8 dataset, the training and testing were performed on each of the eight sensors, with the average results summarized in Table 2 and more detailed results included in the supplementary materials.

Fig. 4. Examples of the images processed using the proposed DMCC method and other methods extracted from [10].

4.2. Impacts of Diagonal Mapping and Feature Extraction

The diagonal mapping and feature extraction are the two key elements in the proposed DMCC method. In order to evaluate which one plays a more important role, we carried out a comparative analysis, with the results summarized in Table 3. The analysis shows that methods like CNN [28] and FC4 [6], as well as DMCC without a diagonal mapping (DMCC (w/o)), all had poor performance. This suggests that feature extraction individual cannot result in good performance. However, when a diagonal mapping is employed, significant improvements can be observed. Specifically, the mean and median errors of the CNN were reduced by 40% and 42% respectively; more significant improvements can be found for the FC4, with a 56% decrease in the mean error and a 64% decrease in the median error. These clearly suggest that the proposed diagonal mapping plays a critical role in solving the cross-sensor color constancy issues.

Table 2. Summary of the performance of various methods, in terms of angular errors, on the Cube+ [26] and NUS-8 [25] datasets. For the NUS-8 dataset, the mean values of the eight sensors are reported here, with the detailed information shown in the supplementary material. The proposed method is highlighted in yellow.

Cube+ Dataset	**Best25%**	**Mean**	**Med.**	**Tri.**	**Worst25%**
Gray-world [3]	0.60	3.52	2.55	2.82	7.98
Shades-of-Gray [27]	0.43	3.22	2.12	2.44	7.77
Quasi-Unsupervised CC [11]	0.49	2.69	1.76	2.00	6.45
SIIE [9]	0.44	2.14	1.44	-	5.06
C5(m=7) [10]	0.41	1.87	1.27	-	4.36
C5(m=1) [10]	0.55	2.60	1.86	-	5.89
DMCC (Ours)	0.49	2.23	1.63	1.78	4.95

NUS-8 Dataset	**Best25%**	**Mean**	**Med.**	**Tri.**	**Worst25%**
Gray-world [3]	1.16	4.59	3.46	3.81	9.85
Shades-of-Gray [27]	0.98	3.67	2.94	3.03	7.75
Cheng-PCA [25]	0.78	2.93	2.33	2.42	6.13
Quasi-Unsupervised CC [11]	-	3.00	2.25	-	-
SIIE [9]	0.52	2.05	1.50	-	4.48
C5(m=7) [10]	0.66	2.68	2.00	-	5.90
C5(m=1) [10]	0.69	2.84	2.20	-	6.14
DMCC (Ours)	0.74	2.80	2.12	2.25	5.88

Table 3. Comparisons of various methods with and without the diagonal mapping and the feature extraction, in terms of the angular error. The model was trained on Canon 5DSR, but tested on Sony IMX135. Sensor-invariant performance was evaluated on Sony IMX135.

Method	Cross-sensor						Sensor-invariant	
	Feature		Diagonal		Dual		-	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
CNN [28](w/o)	6.31	5.25	6.31	5.25	6.31	5.25	3.28	2.28
CNN [28](w)	–	–	**3.77**	**3.07**	–	–	–	–
FC4 [6](w/o)	5.57	4.99	5.57	4.99	5.57	4.99	1.73	1.14
FC4 [6](w)	–	–	**2.46**	**1.80**	–	–	–	–
DMCC(w/o)[a]	10.05	9.10	–	–	–	–	2.55	1.79
DMCC	–	–	–	–	**3.20**	**2.27**	–	–

[a] Roughly equivalent to PCC [7].

Based on this, the combination of the diagonal mapping and the feature extraction in the DMCC method further improves the performance. It can be found that the combination was able to introduce nearly 50% performance improvement over the CNN method without a diagonal mapping, and close to 10% improvement over the CNN with a diagonal mapping, suggesting the benefits brought by the feature extraction. Therefore, it can be concluded that the good performance of the DMCC method was mainly due to the inclusion of diagonal mapping. Though adding the diagonal mapping to the FC4 method can lead to even better performance, it requires significantly more computational resources and memory, as discussed in our previous study [7].

Moreover, the above methods were also evaluated on sensor-invariant conditions, with the results considered as the upper limit for the cross-sensor cases. It is observed that the DMCC method for the cross-sensor cases can reach around 80% of the upper limit (i.e., 3.20 compared to 2.55).

4.3. A Diagonal Matrix or A Full Matrix?

As stated above, the diagonal mapping matrix is derived based on the white points captured by the training and testing sensors under a D65 condition. It's notable that a related work by [29] also employed a diagonal mapping strategy under D55 conditions to compare different sensors and datasets. Although their objective was to identify the biases in color constancy benchmark datasets, it reveals the possibility of using a diagonal mapping when solving color constancy tasks.

It is reasonable to wonder whether a full matrix derived under the conditions with several CCTs (e.g., 2800 and 4000 K), instead of a diagonal matrix, can lead to a better performance. We, therefore, conducted an analysis using the white points captured under three conditions with different CCTs (i.e., 2800, 4000, and 6500 K), with a diagonal matrix derived under each CCT. In addition, a full 3×3 matrix was derived based on the white points captured under three CCTs using a least-square method. As shown in Fig. 5, the diagonal matrix derived under the 6500 K resulted in the best performance, which could be due to the fact that most scenes were under daylight. In contrast, the full matrix did not have a good performance, which should be due to the failure of using a linear transformation to perform a color transformation across different CCTs.

4.4. Further Application, Limitation, and Future Work

The concept of diagonal-matrix mapping, together with the DMCC method, can also be applied for quick evaluations and characterizations of sensor discrepancies. For example, the images captured by Sensor A are considered as the reference, and one of Sensor B and Sensor C needs to be selected so that the captured images can be very similar to the reference. Such a task can be easily performed by adopting the diagonal-matrix mapping, using the white points captured by the three sensors under a D65 condition to transform the images and to calculate the angular errors for selecting either Sensor B or Sensor C.

Last, but not the least, the proposed DMCC method was found to have a poor performance, with an average angular error of 5.5 degrees, on the PolyU Pure Color image dataset [7], a dataset containing 102 images dominated by a single color captured by a HUAWEI P50 Pro smartphone sensor, with the model trained using a Canon 5DSR sensor. This is likely due to the lack of similar pure color images in the training set, and the PolyU Pure Color image dataset only contains the images captured by a single camera. Future work is needed to investigate the performance of cross-sensor methods on corner cases, including pure color images, with more datasets from different sensors to be collected.

5. Conclusion

A DMCC method is proposed in this article for dealing with the cross-sensor illuminant estimation challenge, with a dual-mapping strategy as the key concept. Specifically, the first mapping

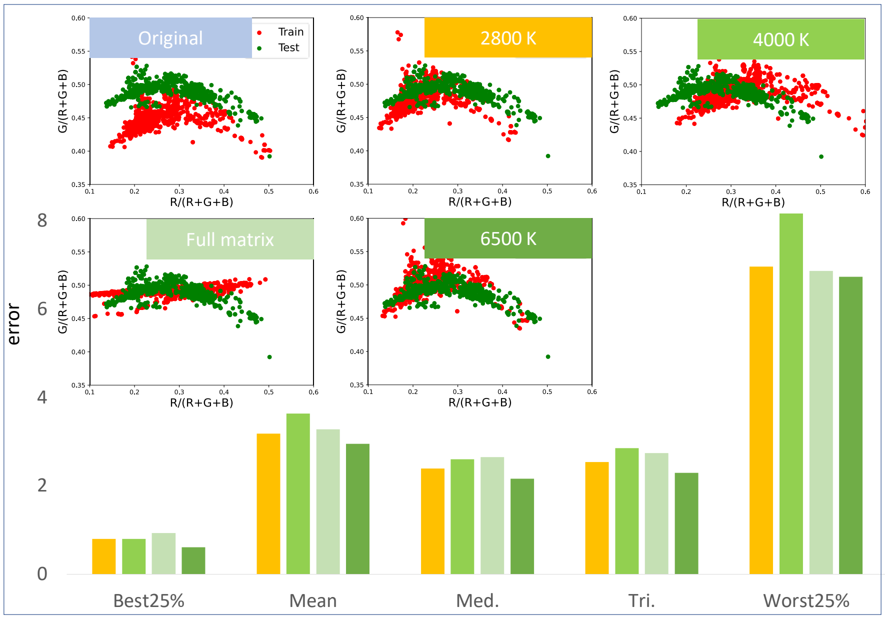

Fig. 5. Illustration of the differences, in terms of the angular error, caused by the different mapping matrices derived using the white points (i.e., a diagonal matrix derived under each of the four CCTs, and a full matrix derived under the four CCTs together).

employs a diagonal matrix, which is derived from the white points captured by the training and testing sensors under a D65 condition, to reconstruct the image data and illuminants. The second mapping then transforms the reconstructed image data into sparse features. These features, together with the reconstructed illuminants, are used to train a lightweight MLP model. The proposed DMCC method was evaluated on three datasets, with the performance being comparable to most of the state-of-the-art methods. Such a good performance comes with a small memory size of ~0.003 MB (1/700 of the state-of-the-art method), allowing for a fast implementation of ~0.3 ms on a GPU (~25 times faster than the state-of-the-art method), and the direct application of the trained model to the testing sensors. This makes the method ready for practical deployment.

Funding. National Science Foundation of China (61975170).

Disclosures. The authors declare no conflicts of interest.

Data availability. No data were generated or analyzed in the presented research. The code of this work can be found at https://github.com/shuwei666/DMCC-Cross-sensor-color-constancy.

Supplemental document. See Supplement 1 for supporting content.

Acknowledgments. Portions of this work were presented at the Color and Imaging Conference in 2023, Practical Cross-sensor Color Constancy using a Dual-mapping Strategy [1].

References

1. S. Yue and M. Wei, “Practical cross-sensor color constancy using a dual-mapping strategy,” in *Color and Imaging Conference,* (2023), pp. 96–101.

2. L. Edwin, “The retinex theory of color vision,” Sci. Am. **237**, 108–128 (1977).
3. G. Buchsbaum, “A spatial processor model for object colour perception,” J. Frankl. Inst. **310**, 1–26 (1980).
4. G. Finlayson and S. Hordley, “Improving gamut mapping color constancy,” IEEE Trans. on Image Process. **9**, 1774–1783 (2000).
5. G. D. Finlayson, “Corrected-moment illuminant estimation,” in *IEEE International Conference on Computer Vision (ICCV),* (2013), pp. 1904–1911.
6. Y. Hu, B. Wang, and S. Lin, “Fc4: Fully convolutional color constancy with confidence-weighted pooling,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2017), pp. 330–339.
7. S. Yue and M. Wei, “Color constancy from a pure color view,” J. Opt. Soc. Am. A. **40**, 602–610 (2023).
8. Y. C. Lo, C. C. Chang, H. C. Chiu, *et al.*, “Clcc: Contrastive learning for color constancy,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2021), pp. 8049–8059.
9. M. Afifi and M. S. Brown, “Sensor-independent illumination estimation for dnn models,” arXiv:1912.06888 (2019).
10. M. Afifi, J. T. Barron, C. LeGendre, *et al.*, “Cross-camera convolutional color constancy,” in *IEEE International Conference on Computer Vision (ICCV),* (2021), pp. 1981–1990.
11. S. Bianco and C. Cusano, “Quasi-unsupervised color constancy,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2019), pp. 12212–12221.
12. J. Xiao, S. Gu, and L. Zhang, “Multi-domain learning for accurate and few-shot color constancy,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2020), pp. 3258–3267.
13. S. McDonagh, S. Parisot, F. Zhou, *et al.*, “Formulating camera-adaptive color constancy as a few-shot meta-learning problem,” arXiv:1811.11788 (2018).
14. D. Ha, A. Dai, and Q. V. Le, “Hypernetworks,” arXiv:1609.09106 (2016).
15. J. T. Barron, “Convolutional color constancy,” in *IEEE International Conference on Computer Vision (ICCV),* (2015), pp. 379–387.
16. J. T. Barron and Y.-T. Tsai, “Fast fourier color constancy,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2017), pp. 886–894.
17. A. Antoniou, H. Edwards, and A. Storkey, “How to train your maml,” arXiv:1810.09502 (2018).
18. D. Hernandez-Juarez, S. Parisot, B. Busam, *et al.*, “A multi-hypothesis approach to color constancy,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2020), pp. 2270–2280.
19. S. Yue and M. Wei, “Dive into illuminant estimation from a pure color viewccf none,” in *Color and Imaging Conference,* (2022), pp. 200–204.
20. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv:1412.6980 (2014).
21. K. He, X. Zhang, S. Ren, *et al.*, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *IEEE International Conference on Computer Vision (ICCV),* (2015), pp. 1026–1034.
22. I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” arXiv:1608.03983 (2016).
23. F. Laakom, J. Raitoharju, J. Nikkanen, A. Iosifidis, and M. Gabbouj, “Intel-tau: A color constancy dataset,” IEEE Access. **9**, 39560–39567 (2021).
24. T. Owens, K. Saenko, A. Chakrabarti, *et al.*, “Learning object color models from multi-view constraints,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2011), pp. 169–176.
25. D. Cheng, D. K. Prasad, and M. S. Brown, “Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution,” J. Opt. Soc. Am. A. **31**, 1049–1058 (2014).
26. N. Banić, K. Koščević, and S. Lončarić, “Unsupervised learning for color constancy,” arXiv:1712.00436 (2017).
27. G. D. Finlayson and E. Trezzi, “Shades of gray and colour constancy,” in *Color and Imaging Conference,* (2004), pp. 37–41.
28. S. Bianco, C. Cusano, and R. Schettini, “Color constancy using cnns,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* (2015), pp. 81–89.
29. M. Buzzelli, S. Zini, S. Bianco, *et al.*, “Analysis of biases in automatic white balance datasets and methods,” Color. Res. & Appl. **48**, 40–62 (2023).