Cross-Camera Convolutional Color Constancy Supplemental Material

Mahmoud Afifi[1,2*] Jonathan T. Barron[1] Chloe LeGendre[1] Yun-Ta Tsai[1] Francois Bleibel[1]

[1]Google Research [2]York University

1. CCC Histogram Features

In the main paper, we used a histogram bin size of 64 (i.e., $n = 64$) with a histogram bin width $\epsilon = (b_{\texttt{max}} - b_{\texttt{min}})/n$, where $b_{\texttt{max}}$ and $b_{\texttt{min}}$ are the histogram boundary values. In our experiments, we set $b_{\texttt{min}}$ and $b_{\texttt{max}}$ to -2.85 and 2.85, respectively. Our input is a concatenation of two histograms: (i) a histogram of pixel intensities and (ii) a histogram of gradient intensities. We augmented our histograms with extra uv coordinate channels to allow our network to consider the "spatial" (or more accurately, chromatic) information associated with each bin in the histogram.

2. Ablations Studies

In the following ablation experiments, we used the Cube+ dataset [8] as our test set and trained our network with seven encoders using the same training set mentioned in the main paper (the NUS dataset [15], the Gehler-Shi dataset [19], and the augmented images after excluding any scene/sensors of the test set). Table 1 shows the results obtained by models trained using different histogram sizes, using different values of the smoothness factors λ_B and λ_F, with and without increasing the batch-size during training, and with and without the histogram gradient intensity and the extra uv augmentation channels. Each experiment was repeated ten times and the arithmetic mean and standard deviation of each error metric are reported.

Figure 1 shows the effect of the smoothness regularization and increasing the batch-size during training on a small training set. We use the first fold of the Gehler-Shi dataset [19] as our validation set and the remaining two folds are used for training. In the figure, we plot the angular error on the training and validation sets. Each model was trained for 60 epochs as a camera-specific color constancy model (i.e., without using additional images or camera models). As can be seen in Figure 1, the smoothness regularization improves the generalization on the test set and increasing the batch size helps the network to reach a lower optimum.

Table 2 shows the results with and without using our data augmentation approach. The experiments labeled "w/aug" in Table 2 refer to using our data augmentation approach, as described in the main paper. Additional details on the data augmentation process are given in Sec. 4.

3. Additional Results

In the main paper, we reported our results using eight additional images. In Table 3, we report multiple versions of our model in which we vary m, the number of input images (and encoders) used ($m = 1$ means that only the query image is used as an input with no additional images). Note that the single-image results ($m = 1$) are not intended to be the central contribution of this work—they are provided *only* as a point of comparison.

We did not include the "gain" multiplier, originally proposed in FFCC [9], in the main paper, as it did not result in a consistent improved performance over all error metrics and datasets. Here, we report results with and without using the gain multiplier map. This gain multiplier map can be generated by our network by adding an additional decoder network with skip connections from the query encoder. Based on this modification, our convolutional structure can now be described as:

$$P = \text{softmax}\left(B + G \circ \sum_i \left(N_i * F_i\right)\right), \qquad (1)$$

where $\{F_i\}$, B, and G are filters, a bias map $B(i,j)$, and the gain multiplier map $G(i,j)$, respectively. We also change the smoothness regularizer to include the generated gain multiplier as follows:

$$\begin{aligned} S\left(\{F_i\}, B, G\right) = \lambda_B(\|B * \nabla_u\|^2 + \|B * \nabla_v\|^2) \\ +\lambda_G(\|G * \nabla_u\|^2 + \|G * \nabla_v\|^2) \\ +\lambda_F \sum_i (\|F_i * \nabla_u\|^2 + \|F_i * \nabla_v\|^2), \end{aligned} \qquad (2)$$

where ∇_u and ∇_v are 3×3 horizontal and vertical Sobel filters, respectively, and λ_F, λ_B, λ_G are scalar multipliers to

*This work was done while Mahmoud was an intern at Google.

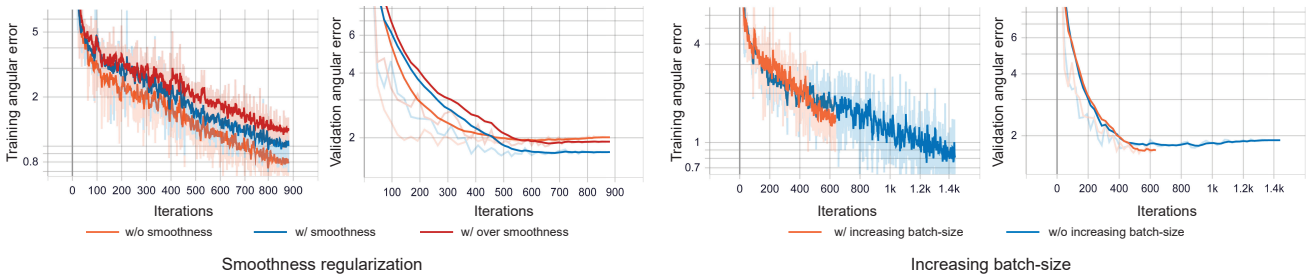


Figure 1: The impact of smoothness regularization and of increasing the batch size during training on training/validation accuracy. We show the training/validation angular error of training our network on the Gehler-Shi dataset [19] for camera-specific color constancy. We set $\lambda_F = 0.15$, $\lambda_B = 0.02$ for the experiment labeled with 'w/ smoothness', while we used $\lambda_F = 1.85$, $\lambda_B = 0.25$ for the experiment labeled with 'over smoothness' and $\lambda_F = 0$, $\lambda_B = 0$ for the 'w/o smoothness' experiments.

Table 1: Results of ablation studies. The shown results were obtained by training our network on the NUS [15] and the Gehler-Shi datasets [19] with augmentation, and testing on the Cube+ dataset [8]. In this set of experiments, we used seven encoders (i.e., six additional histograms). Note that none of the training data includes any scene/sensor from the Cube+ dataset [8]. For each set of experiments, we highlight the lowest errors in yellow.

	Mean	**Med.**	**B. 25%**	**W. 25%**	**Tri.**
Histogram bin size, n					
$n = 16$	2.28±0.01	1.81±0.03	0.65±0.01	4.72±0.02	1.91±0.02
$n = 32$	2.02±0.01	1.44±0.01	0.44±0.01	4.66±0.01	1.86±0.03
$n = 64$	1.87±0.00	1.27±0.01	0.41±0.01	4.36±0.01	1.40±0.01
$n = 128$	2.03±0.00	1.42±0.01	0.40±0.00	4.70±0.01	1.54±0.01
Smoothness factors, λ_B and λ_F ($n = 64$)					
$\lambda_B = 0, \lambda_F = 0$	2.07±0.01	1.42±0.01	0.47±0.01	4.67±0.01	1.57±0.01
$\lambda_B = 0.005, \lambda_F = 0.035$	1.95±0.00	1.31±0.01	0.40±0.00	4.57±0.01	1.47±0.01
$\lambda_B = 0.02, \lambda_F = 0.15$	1.87±0.00	1.27±0.01	0.41±0.01	4.36±0.01	1.40±0.01
$\lambda_B = 0.10, \lambda_F = 0.75$	2.11±0.00	1.55±0.01	0.48±0.00	4.70±0.01	1.66±0.01
$\lambda_B = 0.25, \lambda_F = 1.85$	2.23±0.00	1.61±0.01	0.53±0.00	5.04±0.01	1.77± 0.01
Increasing batch size ($n = 64$)					
w/o increasing	1.93±0.00	1.29±0.01	0.42±0.00	4.52±0.02	1.43±0.01
w/ increasing	1.87±0.00	1.27±0.01	0.41±0.01	4.36±0.01	1.40±0.01
Gradient histogram and uv channels ($n = 64$)					
w/o gradient histogram	2.30±0.01	1.53±0.01	0.45±0.01	5.51±0.02	1.71±0.02
w/o uv	2.03±0.01	1.45±0.01	0.44±0.01	4.63±0.02	1.56±0.01
w/ uv and gradient histogram	1.87±0.00	1.27±0.01	0.41±0.01	4.36±0.01	1.40±0.01

Table 2: Angular errors on the Cube+ dataset [8] and the INTEL-TAU dataset [29]. In this experiment, we used six additional images (i.e., $m = 7$) for our C5. Lowest errors are highlighted in yellow.

Cube+ Dataset	**Mean**	**Med.**	**B. 25%**	**W. 25%**	**Tri.**
Cross-dataset CC [26]	2.47	1.94	-	-	-
Quasi-Unsupervised CC [10]	2.69	1.76	0.49	6.45	2.00
SIIE [3]	2.14	1.44	0.44	5.06	-
FFCC [9]	2.69	1.89	0.46	6.31	2.08
C5	2.10	1.38	0.49	4.97	1.56
C5 (w/aug.)	1.87	1.27	0.41	4.36	1.40

INTEL-TAU	**Mean**	**Med.**	**B. 25%**	**W. 25%**	**Tri.**
Quasi-Unsupervised CC [10]	3.12	2.19	0.60	7.28	2.40
SIIE [3]	3.42	2.42	0.73	7.80	2.64
FFCC [9]	3.42	2.38	0.70	7.96	2.61
C5	2.62	1.85	0.54	6.05	2.00
C5 (w/aug.)	2.49	1.66	0.51	5.93	1.83

control the strength of the smoothness of each of the filters, the bias, and the gain, respectively. The results of using the additional gain multiplier map are reported in Table 4.

We further trained and tested our C5 model using the INTEL-TAU dataset evaluation protocols [29]. Specifically, the INTEL-TAU dataset introduced two different evaluation protocols: (i) the cross-validation protocol, where the model is trained using a 10-fold cross-validation scheme of images taken from three different camera models, and (ii) the camera invariance evaluation protocol, where the model is trained on a single camera model and then tested on another camera model. This camera invariance protocol is equivalent to the CS evaluation method [3], as the models are trained and tested on the same scene set, but with different camera models in the training and testing phases. See Table 5 for comparison with other methods using the INTEL-TAU evaluation protocols. In Table 5, we also show the results of

Table 3: Results using different number of the additional images (i.e., different values of m). Note that $m = 7$, for example, means that we use six additional images along with the input image. For each experiment, we used the same training data explained in the main paper with augmentation. Lowest errors are highlighted in yellow. .

Cube+ Dataset	Mean	Med.	B. 25%	W. 25%
C5 ($m = 1$)	2.60	1.86	0.55	5.89
C5 ($m = 3$)	2.28	1.50	0.59	5.19
C5 ($m = 5$)	2.23	1.52	0.56	5.11
C5 ($m = 7$)	1.87	1.27	0.41	4.36
C5 ($m = 9$)	1.92	1.32	0.44	4.44
C5 ($m = 11$)	1.93	1.41	0.42	4.35
C5 ($m = 13$)	1.95	1.35	0.40	4.52

Cube+ Challenge	Mean	Med.	B. 25%	W. 25%
C5 ($m = 1$)	2.70	2.00	0.61	6.15
C5 ($m = 7$)	2.55	1.63	0.54	6.21
C5 ($m = 9$)	2.24	1.48	0.47	5.39
C5 ($m = 11$)	2.41	1.72	0.54	5.58
C5 ($m = 13$)	2.39	1.61	0.53	5.64

INTEL-TAU	Mean	Med.	B. 25%	W. 25%
C5 ($m = 1$)	2.99	2.18	0.66	6.71
C5 ($m = 7$)	2.49	1.66	0.51	5.93
C5 ($m = 9$)	2.52	1.70	0.52	5.96
C5 ($m = 11$)	2.60	1.79	0.54	6.07
C5 ($m = 13$)	2.57	1.74	0.52	6.08

Gehler-Shi Dataset	Mean	Med.	B. 25%	W. 25%
C5 ($m = 1$)	2.98	2.05	0.54	7.13
C5 ($m = 7$)	2.36	1.61	0.44	5.60
CS ($m = 9$)	2.50	1.99	0.53	5.46
C5 ($m = 11$)	2.55	1.88	0.50	5.77
C5 ($m = 13$)	2.46	1.74	0.50	5.73

NUS Dataset	Mean	Med.	B. 25%	W. 25%
C5 ($m = 1$)	2.84	2.20	0.69	6.14
C5 ($m = 7$)	2.68	2.00	0.66	5.90
CS ($m = 9$)	2.54	1.90	0.61	5.61
C5 ($m = 11$)	2.64	1.99	0.65	5.75
C5 ($m = 13$)	2.49	1.88	0.61	5.43

our C5 model trained on the NUS and Gehler-Shi datasets with augmentation (i.e., our camera-independent model) as reported in the main paper for completeness.

Our C5 model achieves reasonable accuracy when used as a camera-specific model. In this scenario, we trained our model on training images captured by the same test camera model with a single encoder (i.e., $m = 1$). We found that $n = 128$, using the gain multiplier map $G(i, j)$, achieves the best camera-specific results. We report the results of our camera-specific models in Table 6.

Lastly, we show additional qualitative results from the INTEL-TAU dataset [29] in Figure 2. In this figure, we show qualitative examples from our "worst 25%" and "best 25%" results alongside the corresponding results of prior sensor-independent techniques [3, 10].

4. Data Augmentation

In this section, we describe in detail the data augmentation procedure described in the main paper. We begin with the steps used to map a color temperature to the corresponding CIE XYZ value. Then, we elaborate the process of mapping from camera sensor raw to the CIE XYZ color space. Afterwards, we describe the details of the scene retrieval process mentioned in the main paper. Finally, we discuss experiments performed to evaluate our data augmentation and compare it with other color constancy augmentation techniques used in the literature.

4.1. From Color Temperature to CIE XYZ

According to Planck's radiation law [37], the spectral power distribution (SPD) of a blackbody radiator at a given wavelength range $[\lambda, \partial\lambda]$ can be computed using the color temperature q as follows:

$$S_\lambda d_\lambda = \frac{f_1 \lambda^{-5}}{\exp\left(f_2/\lambda q\right) - 1} \partial\lambda, \tag{3}$$

where, $f_1 = 3.741832^1 0^{-16}\ Wm^2$ is the first radiation constant, $f_2 = 1.4388^{10-2} mK$ is the second radiation constant, and q is the blackbody temperature, in Kelvin. [30, 36]. Once the SPD is computed, the corresponding CIE tristimulus values can be approximated in the following discretized form:

$$X = \Delta\lambda \sum_{\lambda=380}^{\lambda=780} x_\lambda S_\lambda, \tag{4}$$

where the value of x_λ is the standard CIE color match value [16]. The values of Y and Z are computed similarly. The corresponding chromaticity coordinates of the computed XYZ tristimulus are finally computed as follows:

$$\begin{aligned} x &= X/(X + Y + Z), \\ y &= Y/(X + Y + Z), \\ z &= Z/(X + Y + Z). \end{aligned} \tag{5}$$

4.2. From Raw to CIE XYZ

Most DSLR cameras provide two pre-calibrated matrices, C_1 and C_2, to map from the camera sensor space to the CIE 1931 XYZ 2-degree standard observer color space. These pre-calibrated color space transformation (CST) matrices are usually provided as a low color temperature (e.g., Standard-A) and a higher correlated color temperature (e.g., D65) [1].

Given an illuminant vector $\boldsymbol{\ell}$, estimated by an illuminant estimation algorithm, the CIE XYZ mapping matrix associated with $\boldsymbol{\ell}$ is computed as follows [14]:

$$C_{T_{\boldsymbol{\ell}}} = C_2 + (1 - \alpha)\, C_1, \tag{6}$$

Table 4: Results of using the gain multiplier, G. For each experiment, we used $m = 7$ and $n = 64$, and trained our network using the same training data explained in the main paper with augmentation. Lowest errors are highlighted in yellow.

	Cube+ [8]				**Cube+ Challenge** [6]				**INTEL-TAU** [29]				**Gehler-Shi** [19]				**NUS** [15]			
	Mean	**Med.**	**B. 25%**	**W. 25%**	**Mean**	**Med.**	**B. 25%**	**W. 25%**	**Mean**	**Med.**	**B. 25%**	**W. 25%**	**Mean**	**Med.**	**B. 25%**	**W. 25%**	**Mean**	**Med.**	**B. 25%**	**W. 25%**
w/o G	1.87	1.27	0.41	4.36	2.40	1.58	0.52	5.76	2.49	1.66	0.51	5.93	2.36	1.61	0.44	5.60	2.68	2.00	0.66	5.90
w/ G	1.83	1.24	0.42	4.25	2.34	1.45	0.46	5.86	2.63	1.81	0.55	6.18	2.36	1.72	0.48	5.40	2.44	1.89	0.64	5.21

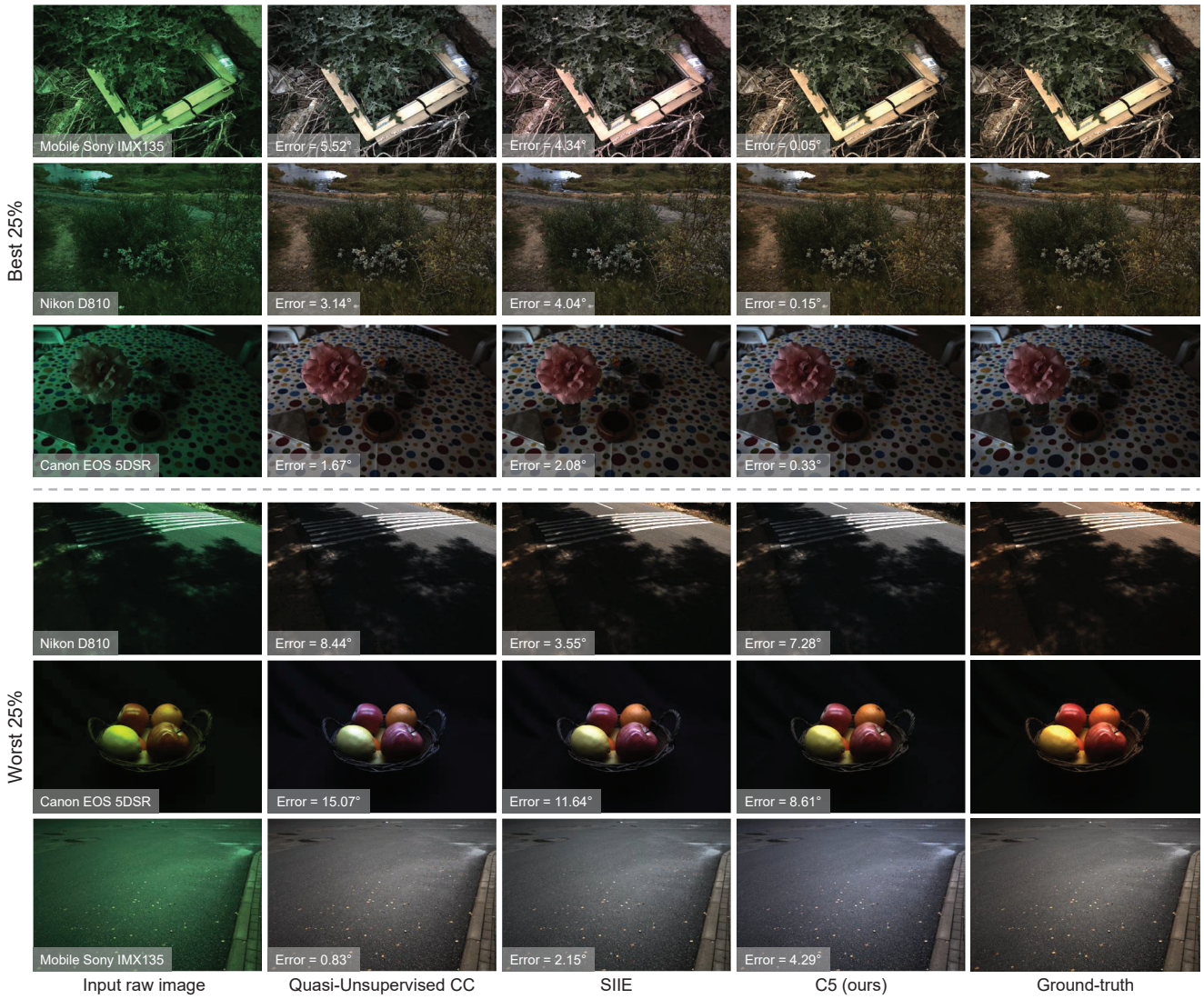


Figure 2: Random examples from our "worst 25%" and "best 25%" results alongside quasi-unsupervised CC [10] and SIIE [3]. Input images are from the INTEL-TAU dataset [29].

$$\alpha = (1/q_{\boldsymbol{\ell}} - 1/q_1)/(1/q_2 - 1/q_1), \tag{7}$$

where q_1 and q_2 are the correlated color temperature associated to the pre-calibrated matrices C_1 and C_2, and $q_{\boldsymbol{\ell}}$ is the color temperature of the illuminant vector $\boldsymbol{\ell}$. Here, $q_{\boldsymbol{\ell}}$ is unknown, and unlike the standard mapping from color temperature to the CIE XYZ space (Sec. 4.1), there is no standard conversion from a camera sensor raw space to the corresponding color temperature. Thus, the conversion from the sensor raw space to the CIE XYZ space is a chicken-and-egg problem—computing the correlated color temperature $q_{\boldsymbol{\ell}}$ is necessarily to get the CST matrix $C_{q_{\boldsymbol{\ell}}}$, while knowing the mapping from a camera sensor raw to the CIE XYZ space inherently requires knowledge of the correlated color temperature of a given raw illuminant.

This problem can be solved by a trial-and-error strategy as follows. We iterate over the color temperature range of 2500K to 7500K. For each color temperature q_i , we first compute the corresponding CST matrix C_{q_i} using Eqs. 6

Table 5: Results using the INTEL-TAU dataset evaluation protocols [29]. We also show the results of camera-independent methods, including our camera-independent C5 model. Lower errors for each evaluation protocol are highlighted in yellow. The best results are bold-faced.

INTEL-TAU [29]	Mean	Med.	B. 25%	W. 25%	Tri.
Camera-specific (10-fold cross-validation protocol [29])					
Bianco et al.'s CNN [11]	3.5	2.6	0.9	7.4	2.8
C3AE [28]	3.4	2.7	0.9	7.0	2.8
BoCF [27]	2.4	1.9	0.7	5.1	2.0
FFCC [9]	2.4	1.6	0.4	5.6	1.8
VGG-FC^4 [22]	**2.2**	1.7	0.6	**4.7**	1.8
C5 ($m=7, n=128$), w/ augmentation	2.33	**1.55**	**0.45**	5.57	**1.71**
Camera-specific (camera invariant protocol [29])					
Bianco et al.'s CNN [11]	3.4	2.5	0.8	7.2	2.7
C3AE [28]	3.4	2.7	0.9	7.0	2.8
BoCF [27]	2.9	2.4	0.9	6.1	2.5
VGG-FC^4 [22]	2.6	2.0	0.7	5.5	2.2
C5 ($m=9$), w/aug.	2.45	1.82	0.53	5.46	1.95
Camera-independent					
Gray-world [13]	4.7	3.7	0.9	10.0	4.0
White-Patch [12]	7.0	5.4	1.1	14.6	6.2
1st-order Gray-Edge [12]	5.3	4.1	1.0	11.7	4.5
2nd-order Gray-Edge [12]	5.1	3.8	1.0	11.3	4.2
Shades-of-Gray [17]	4.0	2.9	0.7	9.0	3.2
PCA-based B/W Colors [15]	4.6	3.4	0.7	10.3	3.7
Weighted Gray-Edge [20]	6.0	4.2	0.9	14.2	4.8
Quasi-Unsupervised CC [10]	3.12	2.19	0.60	7.28	2.40
SIIE [3]	3.42	2.42	0.73	7.80	2.64
C5 ($m=7$), w/aug.	2.49	1.66	0.51	5.93	1.83

Table 6: Results of our C5 trained as a **camera-specific model** with a single encoder (i.e., $m = 1$). In this experiment, we performed a three-fold cross-validation on the Cube+ dataset [8]. For the Cube+ challenge [6], we report our results after training our model on the Cube+ dataset [8] without including any training example from the Cube+ challenge test set [6]. We also show the results of other camera-specific color constancy methods reported in past papers. Lowest angular errors are highlighted in yellow.

Cube+ Dataset [8]	Mean	Med.	B. 25%	W. 25%	Tri.
Color Dog [7]	3.32	1.19	0.22	10.22	-
APAP [5]	2.01	1.36	0.38	4.71	-
Meta-AWB w/ 20 tuning images [32]	1.59	1.02	0.30	3.85	1.15 -
Color Beaver [25]	1.49	0.77	0.21	3.94	-
SqueezeNet-FC^4 [22]	1.35	0.93	0.30	3.24	1.01
FFCC [9]	1.38	0.74	0.19	3.67	0.89
WB-sRGB (modified for raw-RGB) [4]	1.32	0.74	0.18	3.43	-
MDLCC [38]	1.24	0.83	0.26	2.91	0.92
C5 ($n=128$), w/ G	1.39	0.79	0.24	3.55	0.93

Cube+ Challenge [6]	Mean	Med.	B. 25%	W. 25%	Tri.
V Vuk et al., [6]	6.00	1.96	0.99	18.81	2.25
A Savchik et al., [35]	2.05	1.20	0.40	5.24	1.30
Y Qian et al., (1) [34]	2.48	1.56	0.44	6.11	-
Y Qian et al., (2) [34]	2.27	1.26	0.39	6.02	1.35
FFCC [9]	2.1	1.23	0.47	5.38	-
MHCC [21]	1.95	1.16	0.39	4.99	1.25
K Chen et al., [6]	1.84	1.27	0.39	4.41	1.32
WB-sRGB (modified for raw-RGB) [4]	1.83	1.15	0.35	4.60	-
C5 ($n=128$), w/ G	1.72	1.07	0.36	4.27	1.15

and 7. Then, we convert q_i to the corresponding xyz chromaticity triplet using Eqs. 3–5.

Afterwards, we map the xyz chromaticity triplet to the sensor raw space using the following equation:

$$\boldsymbol{\ell}_{\mathtt{raw}(q_i)} = C_{q_i}^{-1} \lambda_{\mathtt{xyz}(q_i)}. \tag{8}$$

We repeated this process for all color temperatures and selected the color temperature/CST matrix that achieves the minimum angular error between $\boldsymbol{\ell}$ and the reconstructed illuminant color in the sensor raw space.

The accuracy of our conversion depends on the pre-calibrated matrices provided by the manufacturer of the DSLR cameras. Other factors that may affect the accuracy of the mapping includes the precision of the standard mapping from color temperature to XYZ space defined by [16], and the discretization process in Eq. 4.

4.3. Raw-to-raw mapping

Here, we describe the details of the mapping mentioned in the main paper. Let A=$\{\mathbf{a}_1, \mathbf{a}_2, ...\}$ represent the "source" set of demosaiced raw images taken by different camera models with the associated capture metadata. Let $T = \{\mathbf{t}_1, \mathbf{t}_2, ...\}$ represent our "target" set of metadata of captured scenes by the target camera model. Here, the capture metadata includes exposure time, aperture size, ISO gain value, and the global scene illuminant color in the camera sensor space. We also assume that we have access to the pre-calibration color space transformation (CST) matrices for each camera model in the sets A and T (available in most DNG files of DSLR images [1]).

Our goal here is to map all raw images in A, taken by different camera models, to the target camera sensor space in T. To that end, we map each image in A to the device-independent CIE XYZ color space [16]. This mapping is performed as follows. We first compute the correlated color temperature, $q^{(i)}$, of the scene illuminant color vector, $\boldsymbol{\ell}^{(i)}_{\mathtt{raw}(A)}$, of each raw image, $I^{(i)}_{\mathtt{raw}(A)}$, in the set A (see Sec. 4.2). Then, we linearly interpolate between the pre-calibrated CST matrices provided with each raw image to compute the final CST mapping matrix, $C_{q^{(i)}}$, [14]. Afterwards, we map each image, $I^{(i)}_{\mathtt{raw}(A)}$, in the set A to the CIE XYZ space. Note that here we represent each image I as matrices of the color triplets (i.e., $I = \{\mathbf{c}^{(k)}\}$), where k is the total number of pixels in the image I. We map each raw image to the CIE XYZ space as follows:

$$I^{(i)}_{\mathtt{xyz}(A)} = C_{q^{(i)}} D_{\boldsymbol{\ell}^{(i)}} I^{(i)}_{\mathtt{raw}(A)}, \tag{9}$$

where $D_{\boldsymbol{\ell}^{(i)}}$ is the white-balance diagonal correction matrix constructed based on the illuminant vector $\boldsymbol{\ell}^{(i)}_{\mathtt{raw}(A)}$.

Similarly, we compute the inverse mapping from the CIE XYZ space back to the target camera sensor space based on the illuminant vectors and pre-calibration matrices provided in the target set T. The mapping from the source sensor space to the target one in T can be performed as follows:

$$I^{(i)}_{\mathtt{raw}(T)} = D^{-1}_{j^{(i)}} M^{-1}_{q^{(i)}} I^{(i)}_{\mathtt{xyz}(A)}, \tag{10}$$

where $\jmath^{(i)}_{\texttt{raw}(T)}$ is the corresponding illuminant color to the correlated color temperature, $q^{(i)}$, in the target sensor space (i.e., the ground-truth illuminant for image $I^{(i)}_{\texttt{raw}(T)}$ in the illuminant estimation task), and $M^{-1}_{q^{(i)}}$ is the CST matrix that maps from the target sensor space to the CIE XYZ space.

The described steps so far assume that the spectral sensitivities of all sensors in A and T satisfy the Luther condition [33]. Prior studies, however, showed that this assumption is not always satisfied, and this can affect the accuracy of the pre-calibration matrices [23, 24]. According to this, we rely on Eqs. 9 and 10 only to map the original colors of captured objects in the scene (i.e., white-balanced colors) to the target camera model. For the values of the global color cast, $\jmath^{(i)}_{\texttt{raw}(T)}$, we do not rely on $M^{-1}_{q^{(i)}}$ to map $\boldsymbol{\ell}^{(i)}_{\texttt{raw}(A)}$ to the target sensor space of T. Instead, we follow a K-nearest neighbor strategy to get samples from the target sensor's illuminant color space.

4.4. Scene Sampling

As described in the paper, we retrieve metadata of similar scenes in the target set T for illuminant color sampling. This sampling process should consider the source scene capture conditions to sample suitable illuminant colors from the target camera model space—i.e., having indoor illuminant colors as ground-truth for outdoor scenes may affect the training process. To this end, we introduce a retrieval feature $v^{(i)}_A$ to represent the capture settings of the image $I^{(i)}_{\texttt{raw}(A)}$. This feature includes the correlated color temperature and auxiliary capture settings. These additional capture settings are used to retrieve scenes captured with similar settings of $I^{(i)}_{\texttt{raw}(A)}$.

Our feature vector is defined as follows:

$$v^{(i)}_A = [q^{(i)}_{\texttt{norm}}, h^{(i)}_{\texttt{norm}}, p^{(i)}_{\texttt{norm}}, e^{(i)}_{\texttt{norm}}], \tag{11}$$

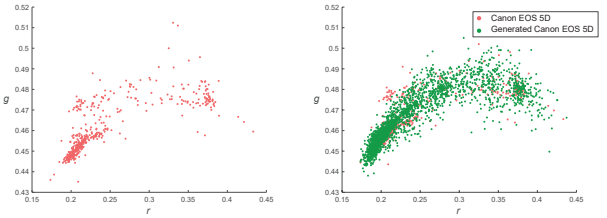
where $q^{(i)}_{\texttt{norm}}$, $h^{(i)}_{\texttt{norm}}$, $p^{(i)}_{\texttt{norm}}$, and $e^{(i)}_{\texttt{norm}}$ are the normalized color temperature, gain value, aperture size, and scaled exposure time, respectively. The gain value and the scaled exposure time are computed as follows:

$$h^{(i)} = \texttt{BLN}^{(i)}\,\texttt{ISO}^{(i)}\,, \tag{12}$$

$$e^{(i)} = \sqrt{2^{\texttt{BLE}^{(i)}}}\,l^{(i)}\,, \tag{13}$$

where BLE, BLN, ISO, and l are the baseline exposure, baseline noise, digital gain value, and exposure time (in seconds), respectively.

Illuminant Color Sampling A naive sampling from the associated illuminant colors in T does not introduce new illuminant colors over the Planckian locus of the target sensor. For this reason, we first fit a cubic polynomial to the



Real illuminant *rg* color distribution taken by Canon EOS 5D

Generated illuminants follow the distribution of the real sensor's illuminant colors

Figure 3: Synthetic illuminant samples of Canon EOS 5D camera model in the Gehler-Shi dataset [19]. The shown generated illuminant colors are then applied to sensor-mapped raw images, originally were taken by different camera models, for augmentation purpose (Sec. 4).

rg chromaticity of illuminant colors in the target sensor T. Then, we compute a new r chromaticity value for each query vector as follows:

$$r_v = \sum_{j \in K} w_j r_j + x\,, \tag{14}$$

where $w_j = \exp(1-d_j)/\sum_k^K \exp(1-d_k)$ is a weighting factor, $x = \lambda_r \mathcal{N}(0, \sigma_r)$ is a small random shift, λ_r is a scalar factor to control the amount of divergence from the ideal Planckian curve, σ_r is the standard deviation of the r chromaticity values in the retrieved K metadata of the target camera model, T_K, and d_j is the normalized L2 distance between $v_{S(i)}$ and the corresponding $j^{\texttt{th}}$ feature vector in T_K. The CST matrix M (Eq. 10) is constructed by linearly interpolating between the corresponding CST matrices associated with each sample in T_K using w_j. After computing r_v, the corresponding g chromaticity value is computed as:

$$g_v = [r_v, r_v^2, r_v^3][\xi_1, \xi_2, \xi_3]^\top + y\,, \tag{15}$$

where $[\xi_1, \xi_2, \xi_3]$ are the cubic polynomial coefficients, y is a random shift, and σ_g is the standard deviation of the g chromaticity values in T_K. In our experiments, we set $\lambda_r = 0.7$ and $\lambda_g = 1$. The final illuminant color $\jmath^{(i)}_{\texttt{raw}(T)}$ can be represented as follows:

$$\jmath^{(i)}_{\texttt{raw}(T)} = [r_v, g_v, 1 - r_v - g_v]^\top\,. \tag{16}$$

To avoid any bias towards the dominant color temperature in the source set, A, we first divide the color temperature range of the source set A into different groups with a step of 250K. Then, we uniformly sample examples from each group to avoid any bias towards specific type of illuminants. Figure 3 shows examples of the sampling process. As shown, the sampled illuminant chromaticity values follow the original distribution over the Planckian curve, while introducing new illuminant colors of the target sensors that

Figure 4: Example of camera augmentation used to train our network. The shown left raw image is captured by Nikon D5200 camera [15]. The next three images are the results of our mapping to different camera models.

were not included in the original set. Finally, we apply random cropping to introduce more diversity in the generated images. Figure 4 shows examples of synthetic raw-like images of different target camera models.

4.5. Evaluation

In prior work, several approaches for training data augmentation for illuminant estimation have been attempted [2, 18, 31]. These approaches first white-balance the training raw images using the associated ground-truth illuminant colors associated with each image. Afterwards, illuminant colors are sampled from the "ground-truth" illuminant colors over the entire training set to be applied to the white-balanced raw images. These sampled illuminant colors can be taken randomly from the ground-truth illuminant colors [18] or after clustering the ground-truth illuminant colors [31]. These methods, however, are limited to using the same set of scenes as is present in the training dataset. Another approach for data augmentation has been proposed in [2] by mapping sRGB white-balanced images to a learned normalization space that is is learned based on the CIE XYZ space. Afterwards, a pre-computed global transformation matrix is used to map the images from this normalization space to the target white-balanced raw space. In contrast, the augmentation method described in our paper uses an accurate mapping from the camera sensor raw space to the CIE XYZ using the pre-calibration matrices provided by camera manufacturers.

In the following set of experiments, we use the baseline model FFCC [9] to study the potential improvement of our chosen data augmentation strategy and alternative augmentation techniques proposed in [2, 18, 31]. We use the Canon EOS 5D images from in the Gehler-Shi dataset [19] for comparisons. For our test set, we randomly select 30% of the total number of images in the Canon EOS 5D set. The remaining 70% of images are used for training. We refer to this set as "real training set", which includes 336 raw images.

Note that, except for the augmentation used in a [2], none of these methods apply a sensor-to-sensor mapping, as they use the raw images of the "real training set" as the source and target set for augmentation. For this reason and for a fair comparison, we provide the results of two different set of experiments. In the first experiment, we use the CIE XYZ images taken by the Canon EOS 5D sensor as our source set A, while in the second experiment, we use a different set of four sensors rather than the Canon EOS 5D sensor. The former is comparable to the augmentation methods used in [18, 31] (see Table 7), while the latter is comparable to the augmentation approach used in [2], which performs "raw mapping" in order to introduce new scene content in the training data (see Table 8). The shown results obtained by generating 500 synthetic images by each augmentation method, including our augmentation approach. As shown in Tables 7 and 8, our augmentation approach achieves the best improvement of the FFCC results.

In order to study the effect of the CIE XYZ mapping used by our augmentation approach, we trained FFCC [9] on a set of 500 synthetic raw images of the target camera model—namely, the Canon EOS 5D camera model in the Gehler-Shi dataset [19]. These synthetic raw images were originally captured by the Canon EOS 1Ds Mark III camera sensor (in the NUS dataset [15]), then these images are mapped to the target sensor using our augmentation approach. Table 9 shows the results of FFCC trained on synthetic raw images with and without the intermediate CIE XYZ mapping step (Eqs. 9 and 10). As shown, using the CIE XYZ mapping achieves better results, which are further improved by increasing the scene diversity of the source set by including additional scenes from other datasets, as shown in Table 8.

For a further evaluation, we use our approach to map images from the Canon EOS 5D camera's set (the same set that was used to train the FFCC model) to different target camera models. Then, we trained and tested a FFCC model on these mapped images. This experiment was performed to gauge the ability of our data augmentation approach to have similar negative effects on camera-specific methods that were trained on a different camera model. To that end, we randomly selected 150 images from the Canon EOS 5D sensor set, which was used to train the FFCC model, as our source image set A. Then, we mapped these images to different target camera models using our approach. That means that the training and our synthetic testing set share the same scene content. We report the results in Table 10. We also report the testing results on real image sets captured by the same target camera models. As shown in Table 10, both real and synthetic sets negatively affect the accuracy of the FFCC model (see Table 7 for results of the FFCC on a testing set taken by the same training sensor).

Table 7: A comparison of different augmentation methods for illuminant estimation. All results were obtained by using training images captured by the Canon EOS 5D camera model [19] as the source and target sets for augmentation. Lowest errors are highlighted in yellow.

Training set	Mean	Med.	B. 25%	W. 25%
Original set	1.81	1.12	0.35	4.43
Augmented (clustering & sampling) [31]	1.68	0.97	0.25	4.31
Augmented (sampling) [18]	1.79	1.09	0.33	4.34
Augmented (ours)	1.55	0.98	0.28	3.68

Table 8: A comparison of techniques for generating new sensor-mapped raw-like images that were originally captured by different sensors than the training camera model. The term 'synthetic' refers to training FFCC [9] without including any of the original training examples, while the term 'augmented' refers to training on synthetic and real images. The best results are bold-faced. Lowest errors of synthesized and augmented sets are highlighted in red and yellow, respectively.

Training set	Mean	Med.	B. 25%	W. 25%
Synthetic [2]	4.17	3.06	0.78	9.39
Augmentation [2]	2.64	1.95	0.45	5.97
Synthetic (ours)	2.44	1.89	0.42	5.40
Augmented (ours)	**1.75**	**1.28**	**0.35**	**4.15**

Table 9: Results of FFCC [9] trained on synthetic raw-like images after they are mapped to the target camera model. In this experiment, the raw images are mapped from the Canon EOS-1Ds Mark III camera sensor (taken from the NUS dataset [15]) to the target Canon EOS 5D camera in the Gehler-Shi dataset [19]. The shown results were obtained with and without the intermediate CIE XYZ mapping step to generate the synthetic training set. Lowest errors are highlighted in yellow.

Synthetic training set	Mean	Med.	B. 25%	W. 25%
w/o CIE XYZ	3.30	2.55	0.60	7.21
w/ CIE XYZ	3.04	2.36	0.56	6.58

Table 10: Results of FFCC [9] trained on the Canon EOS 5D camera [19] and tested on images taken by different camera models from the NUS dataset [15] and the Cube+ challenge set [8]. The synthetic sets refer to testing images generated by our data augmentation approach, where these images were mapped from the Canon EOS 5D set (used for training) to the target camera models.

Testing sensor	Real camera images			Synthetic camera images		
	Mean	Med.	Max	Mean	Med.	Max
Canon EOS 1D [19]	3.88	2.66	16.32	4.68	3.80	22.83
Fujifilm XM1 [15]	4.22	3.05	47.87	2.91	2.06	38.93
Nikon D5200 [15]	4.45	3.45	36.762	3.36	2.10	41.23
Olympus EPL6 [15]	4.35	3.56	19.89	3.28	2.27	38.81
Panasonic GX1 [15]	2.83	2.03	16.58	3.24	2.29	17.07
Samsung NX2000 [15]	4.41	3.73	17.69	3.44	2.64	18.79
Sony A57 [15]	3.84	3.02	19.38	3.04	1.34	39.67
Canon EOS 550D [8]	3.83	2.49	46.55	3.14	1.98	36.30

References

[1] Digital negative (DNG) specification. Technical report, Adobe Systems Incorporated, 2012. Version 1.4.0.0.

[2] Mahmoud Afifi, Abdelrahman Abdelhamed, Abdullah Abuolaim, Abhijith Punnappurath, and Michael S Brown. CIE XYZ Net: Unprocessing images for low-level computer vision tasks. *arXiv preprint arXiv:2006.12709*, 2020.

[3] Mahmoud Afifi and Michael S Brown. Sensor-independent illumination estimation for dnn models. *BMVC*, 2019.

[4] Mahmoud Afifi, Brian Price, Scott Cohen, and Michael S Brown. When color constancy goes wrong: Correcting improperly white-balanced images. *CVPR*, 2019.

[5] Mahmoud Afifi, Abhijith Punnappurath, Graham Finlayson, and Michael S. Brown. As-projective-as-possible bias correction for illumination estimation algorithms. *JOSA A*, 2019.

[6] Nikola Banić and Karlo Koščević. Illumination estimation challenge. `https://www.isispa.org/illumination-estimation-challenge`. Accessed: 2021-03-07.

[7] Nikola Banic and Sven Loncaric. Color dog-guiding the global illumination estimation to better accuracy. *VISAPP*, 2015.

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

[9] Jonathan T Barron and Yun-Ta Tsai. Fast Fourier color constancy. *CVPR*, 2017.

[10] Simone Bianco and Claudio Cusano. Quasi-Unsupervised color constancy. *CVPR*, 2019.

[11] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using cnns. *CVPR Workshops*, 2015.

[12] David H Brainard and Brian A Wandell. Analysis of the retinex theory of color vision. *JOSA A*, 1986.

[13] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 1980.

[14] Hakki Can Karaimer and Michael S Brown. Improving color reproduction accuracy on cameras. *CVPR*, 2018.

[15] 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. *JOSA A*, 2014.

[16] C CIE. Commission internationale de l'eclairage proceedings, 1931. *Cambridge University, Cambridge*, 1932.

[17] Graham D Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. *Color and Imaging Conference*, 2004.

[18] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Trémeau, and Christian Wolf. Mixed pooling neural networks for color constancy. *ICIP*, 2016.

[19] Peter V Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. *CVPR*, 2008.

[20] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Improving color constancy by photometric edge weighting. *TPAMI*, 2012.

[21] Daniel Hernandez-Juarez, Sarah Parisot, Benjamin Busam, Ales Leonardis, Gregory Slabaugh, and Steven McDonagh.

A multi-hypothesis approach to color constancy. *CVPR*, 2020.

[22] Yuanming Hu, Baoyuan Wang, and Stephen Lin. FC4: Fully convolutional color constancy with confidence-weighted pooling. *CVPR*, 2017.

[23] Jun Jiang, Dengyu Liu, Jinwei Gu, and Sabine Süsstrunk. What is the space of spectral sensitivity functions for digital color cameras? *WACV*, 2013.

[24] Hakki Can Karaimer and Michael S Brown. Beyond raw-RGB and sRGB: Advocating access to a colorimetric image state. *Color and Imaging Conference*, 2019.

[25] Karlo Koščević, Nikola Banić, and Sven Lončarić. Color beaver: Bounding illumination estimations for higher accuracy. *VISIGRAPP*, 2019.

[26] Samu Koskinen12, Dan Yang, and Joni-Kristian Kämäräinen. Cross-dataset color constancy revisited using sensor-to-sensor transfer. *BMVC*, 2020.

[27] Firas Laakom, Nikolaos Passalis, Jenni Raitoharju, Jarno Nikkanen, Anastasios Tefas, Alexandros Iosifidis, and Moncef Gabbouj. Bag of color features for color constancy. *IEEE TIP*, 2020.

[28] Firas Laakom, Jenni Raitoharju, Alexandros Iosifidis, Jarno Nikkanen, and Moncef Gabbouj. Color constancy convolutional autoencoder. *Symposium Series on Computational Intelligence*, 2019.

[29] Firas Laakom, Jenni Raitoharju, Alexandros Iosifidis, Jarno Nikkanen, and Moncef Gabbouj. Intel-TAU: A color constancy dataset. *arXiv preprint arXiv:1910.10404*, 2019.

[30] Changjun Li, Guihua Cui, Manuel Melgosa, Xiukai Ruan, Yaoju Zhang, Long Ma, Kaida Xiao, and M Ronnier Luo. Accurate method for computing correlated color temperature. *Optics express*, 2016.

[31] Zhongyu Lou, Theo Gevers, Ninghang Hu, Marcel P Lucassen, et al. Color constancy by deep learning. *BMVC*, 2015.

[32] Steven McDonagh, Sarah Parisot, Fengwei Zhou, Xing Zhang, Ales Leonardis, Zhenguo Li, and Gregory Slabaugh. Formulating camera-adaptive color constancy as a few-shot meta-learning problem. *arXiv preprint arXiv:1811.11788*, 2018.

[33] Junichi Nakamura. *Image sensors and signal processing for digital still cameras*. CRC press, 2017.

[34] Yanlin Qian, Ke Chen, and Huanglin Yu. Fast fourier color constancy and grayness index for ISPA illumination estimation challenge. *International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019.

[35] A Savchik, E Ershov, and S Karpenko. Color cerberus. *International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019.

[36] John Walker. Colour rendering of spectra. `https://www.fourmilab.ch/documents/specrend/`. Accessed: 2021-03-07.

[37] Gunter Wyszecki and Walter Stanley Stiles. *Color science*, volume 8. Wiley New York, 1982.

[38] Jin Xiao, Shuhang Gu, and Lei Zhang. Multi-domain learning for accurate and few-shot color constancy. *CVPR*, 2020.