Leveraging the Availability of Two Cameras for Illuminant Estimation

Abdelrahman Abdelhamed Abhijith Punnappurath Michael S. Brown
Samsung AI Center – Toronto
{a.abdelhamed,abhijith.p,michael.b1}@samsung.com

Abstract

Most modern smartphones are now equipped with two rear-facing cameras – a main camera for standard imaging and an additional camera to provide wide-angle or telephoto zoom capabilities. In this paper, we leverage the availability of these two cameras for the task of illumination estimation using a small neural network to perform the illumination prediction. Specifically, if the two cameras' sensors have different spectral sensitivities, the two images provide different spectral measurements of the physical scene. A linear 3×3 color transform that maps between these two observations – and that is unique to a given scene illuminant – can be used to train a lightweight neural network comprising no more than 1460 parameters to predict the scene illumination. We demonstrate that this two-camera approach with a lightweight network provides results on par or better than much more complicated illuminant estimation methods operating on a single image. We validate our method's effectiveness through extensive experiments on radiometric data, a quasi-real two-camera dataset we generated from an existing single camera dataset, as well as a new real image dataset that we captured using a smartphone with two rear-facing cameras.

1. Introduction

An overwhelming percentage of consumer photographs are currently captured using smartphone cameras. A recent trend in smartphone imaging system design is to employ two (or more) rear-facing cameras to ameliorate the limitations imposed by the smartphone compact form factor. In most cases, the two rear-facing cameras have different focal lengths and lens configurations to allow the smartphone to deliver DSLR-like optical capabilities (i.e., wide-angle and telephoto). In addition, the two-camera setup has been leveraged for applications such as synthetic bokeh effect [48] and reflection removal [40]. Given the utility of the two-camera configuration, this design trend is likely to continue for the foreseeable future. In this work, we show that the two-camera setup has another benefit, that of improving the accuracy of illuminant estimation.

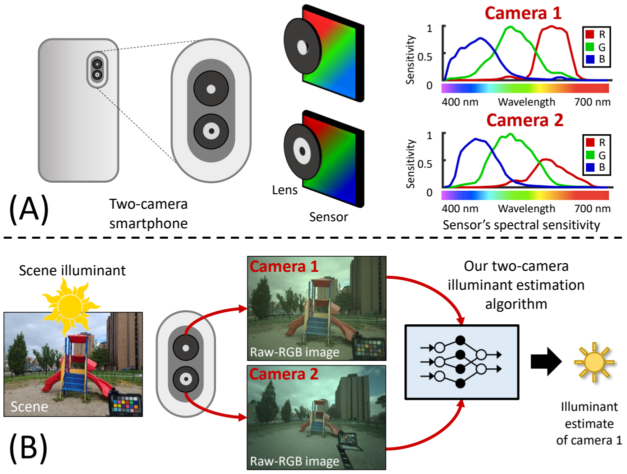


Figure 1: (A) Most modern smartphones use two rear-facing cameras. Typically, the spectral characteristics of these two cameras' sensors are slightly different. (B) Thus, a two-camera system furnishes two different measurements of the scene being imaged. Our proposed two-camera algorithm harnesses this extra information for more accurate and efficient illuminant estimation.

Illuminant estimation is the most critical step for computational color constancy. Color constancy refers to the ability of the human visual system to perceive scene colors as being the same even when observed under different illuminations [39]. Cameras do not innately possess this illumination adaptation ability; the raw-RGB image recorded by the camera sensor has significant color cast due to the scene's illumination. As a result, *computational* color constancy is applied to the camera's raw-RGB sensor image as one of the first steps in the in-camera imaging pipeline to remove this undesirable color cast. The main goal of the camera's auto-white-balance (AWB) module, which is motivated by the concept of computational color constancy, is illuminant estimation. AWB involves estimating the scene illumination in the sensor's raw-RGB color space and then applying a simple 3×3 diagonal matrix computed directly from the estimated illumination parameters to perform the white-balance correction. Thus, accurate estimation of the scene illumination is crucial to ensuring correct scene colors in the camera image.

We demonstrate that two-camera systems have the potential to provide more accurate illuminant estimation compared to existing single-camera methods. A key insight is that the spectral characteristics of the main camera's sensor are typically different from that of the second camera's. This is due to a variety of reasons. For example, the pitch of the photodiodes and overall resolution of the two sensors are often different to accommodate the different optics associated with each sensor. These differences impact which color filter arrays (CFA) manufacturers can use in the sensor's production process. This results in the two CFAs having different spectral sensitivities to incoming light. While on the surface this may appear to be a disadvantage, differences in the CFA between the two cameras can be corrected for by the later stages of the camera imaging pipeline to ensure the final output colors appear the same (e.g., see [36]). However, for our purpose, the sensors' unprocessed raw images effectively provide *different* spectral measurements of the underlying scene. It is this complementary information that allows us to design a two-camera illumination estimation algorithm as shown in Fig. 1.

Contribution We propose to train a neural network for illuminant estimation that receives as input a 3×3 matrix computed between the two cameras' raw sensor images simultaneously capturing the same scene. Prior work [21] has shown that the color transformation between different spectral samples of the same scene has a unique signature that is related to the scene illumination. This allows the color transformation itself to be used as the feature for illumination estimation. Thus, in contrast to existing single-camera illumination estimation methods that train their deep networks directly on image data, or on image histograms, our network needs to examine only nine parameters in the color transformation matrix. As a result, we can train a very lightweight neural network comprising just 1460 parameters that can be efficiently run on-device in real time. We test our proposed approach extensively with experiments on radiometric data, a quasi-real two-camera dataset we generated from an existing single-camera color constancy dataset [16], and finally on a real two-camera dataset that we captured using a Samsung S20 Ultra smartphone. We compare our technique against several state-of-the-art single-image illuminant estimation methods and demonstrate on par or even improved performance.

2. Related work

We survey works on computational color constancy. These algorithms can be broadly categorized into (1) statistics-based and (2) learning-based methods. While early learning-based approaches used hand-crafted features, more recent works employ deep neural networks.

Statistics-based methods operate using statistics from an image's color distribution and spatial layout to estimate the scene illuminant. Representative examples include gray world [15], general gray world [6], gray edges [46], shades of gray [24], white patch [14], bright pixels [35], and PCA [16]. These methods are fast and easy to implement; however, they make very strong assumptions about scene content and fail in cases where these assumptions do not hold.

Learning-based methods use labelled training data where the ground truth illumination corresponding to each input image is known from physical color charts placed in the scene. In general, learning-based approaches are shown to be more accurate than statistical-based methods. However, learning-based methods usually include many more parameters than statistics-based ones; their number could reach up to tens of millions in some models (e.g., [10]) and they typically have relatively longer training time. Representative learning-based approaches include Bayesian methods [13, 26, 44], gamut-based methods [23, 25, 28], exemplar-based methods [5, 27, 34], and bias-correction methods [4, 18, 19]. While early learning-based methods used hand-crafted features, more recently, deep neural networks (DNN) have demonstrated superior performance [32, 38, 41, 45, 11, 12, 43, 10, 31, 47, 3, 8, 9]. It is important to note that the aforementioned methods are designed to work with a single image captured using three channel sensor. Work by [42] explored the idea of adding an additional color channel, but in the context of resolving scene metamerism. The approach in this paper is based on a pair of images of the same scene captured using a two-camera system.

Our approach is inspired by the chromagenic color constancy technique of Finlayson et al. [22, 21, 17]. The chromagenic approach showed that the parameters of a 3×3 linear transform that relates the color values of a scene captured with different spectral sensitivities are correlated with the scene's illumination. The chromagenic approach used two images captured from the same sensor, but with a color filter applied between image capture; however, two sensors with different spectral sensitivities could also be used. Classification of the scene illumination was performed using a set of pre-selected illuminants via a nearest-neighbour search operation. We build on this method and integrate it into a modern smartphone design with two cameras with different fields of view. Furthermore, we combine it with the power of neural networks to regress over the space of illuminations.

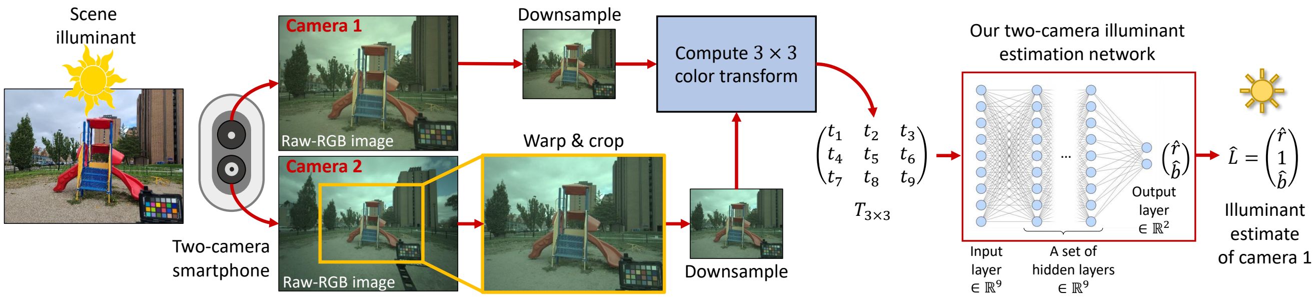

Scene illuminant
Camera 1
Raw-RGB image
Camera 2
Raw-RGB image
Two-camera smartphone
Downsample
Warp & crop
Compute 3 × 3 color transform
Downsample
$\begin{pmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & t_9 \end{pmatrix}$
$T_{3\times3}$
Our two-camera illuminant estimation network
Input layer $\in \mathbb{R}^9$
A set of hidden layers $\in \mathbb{R}^9$
Output layer $\in \mathbb{R}^2$
$\begin{pmatrix} \hat{r} \\ \hat{b} \end{pmatrix}$
$\hat{L} = \begin{pmatrix} \hat{r} \\ 1 \\ \hat{b} \end{pmatrix}$
Illuminant estimate of camera 1

Figure 2: An overview of our proposed two-camera illuminant estimation algorithm. We compute a linear 3×3 transform matrix T that maps the downsampled raw-RGB image from the main camera to the corresponding aligned and downsampled raw-RGB image from the second camera. For a particular scene illuminant, this color transformation T is unique [21]. We feed this mapping T as input to a small lightweight neural network. The network predicts a 2D [R/G B/G] chromaticity value that corresponds to the illuminant estimate of the main camera.

3. Two-camera illuminant estimation

In this section, we describe the various steps of our two-camera illuminant estimation algorithm – spatially aligning the image pairs (Section 3.1); computing color transforms between them (Section 3.2); constructing our two-camera illuminant estimation network (Section 3.3); and augmenting our training data (Section 3.4).

3.1. Image spatial alignment

Our method is based on computing a color transform between a pair of images captured using a two-camera system. These two images usually have different views and need to be registered before computing the color transform. In our experiments on real images, we found that a global homography is sufficient for image alignment. We downsample the images by a factor of six prior to computing the color transform, and this makes our method robust to any small misalignments and slight parallax in the two views. Moreover, since the hardware arrangement of the two cameras does not change for a given device, the homography can be pre-computed and remains fixed for all image pairs from the same device.

3.2. Color transforms for image pairs

Given two raw-RGB images $I_1 \in \mathbb{R}^{n\times3}$ and $I_2 \in \mathbb{R}^{n\times3}$ with n pixels of the same scene captured by two different sensors or cameras, under the same illumination $L \in \mathbb{R}^3$, there exists a linear transformation $T \in \mathbb{R}^{3\times3}$ between the color values of the two images as

$$I_2 \approx I_1\, T, \tag{1}$$

such that T is unique to the scene illumination L [22, 21]. Despite Equation 1 being an approximation, for simplicity, we will use the equality sign instead. We first spatially align the two images using the pre-computed homography, downsample them, and then compute T using the pseudo inverse as follows:

$$T = ({I_1}^T\, I_1)^{-1}\, {I_1}^T\, I_2. \tag{2}$$

3.3. Two-camera illuminant estimation network

Given a dataset of M image pairs

$$\mathcal{I} = \{(I_{11}, I_{21}), \ldots, (I_{1M}, I_{2M})\}, \tag{3}$$

we compute the corresponding color transformations between each pair of images using Equation 2:

$$\mathcal{T} = \{T_1, \ldots, T_M\}. \tag{4}$$

Given the set of corresponding target ground truth illuminants of I_{1i} (i.e., as measured by the first camera) from each pair

$$\mathcal{L} = \{L_1, \ldots, L_M\}, \tag{5}$$

we can train a neural network $f_\theta : \mathcal{T} \rightarrow \mathcal{L}$, with parameters θ, to model the mapping between the color transforms $\mathcal{T}$ and scene illuminations $\mathcal{L}$. Then, f_θ can be used to predict the scene illumination for the main camera given the color transform between the two images

$$\hat{L} = f_\theta\,(T)\,. \tag{6}$$

Without loss of generality, our method can be trained to predict the illuminant for the second camera as well, using the same color transforms; however, for simplicity, we focus on estimating the illumination for the main camera only. We train our network by minimizing the L_1 loss between the predicted illuminants and the ground truth:

$$\min_\theta \frac{1}{M} \sum_{i=1}^{M} \left| \hat{L}_i - L_i \right|. \tag{7}$$

Our network of choice is lightweight, consisting of a small number (e.g., 2, 5, or 16) of dense layers; each layer has nine neurons only. The total number of parameters

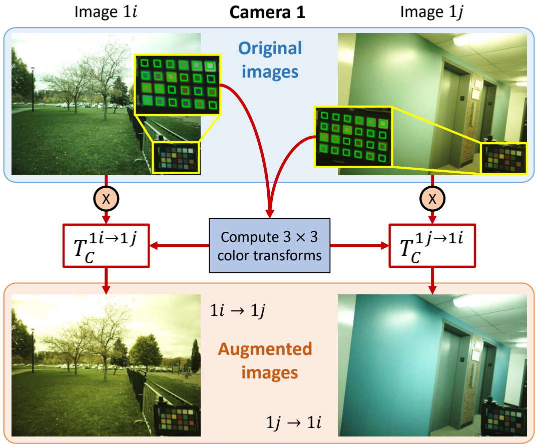


Figure 3: Our image illumination augmentation method. Given a pair of images, we re-illuminate them by each other's illumination based on 3×3 color transformations between their color chart values. This figure shows augmentation of an image pair from one camera only. The corresponding pair of images from the second camera is augmented in the same way. The images shown are in demosaiced raw-RGB format with gamma correction for better visualization.

ranges from 200 for the 2-layer architecture up to 1460 parameters for the 16-layer network. The input to the network is the flattened nine values of the color transform T and the output is two values corresponding to the illumination estimation in the 2D [R/G B/G] chromaticity color space where the green channel's value is always set to 1. An overview of our method is provided in Fig. 2.

3.4. Data augmentation

Due to the lack of large datasets of image pairs captured with two cameras under the same illumination, and to increase the number of training samples and the generalizability of our model, we propose to augment the training images as follows. Given a small dataset of raw-RGB image pairs captured with two cameras and including color rendition charts, we extract the color values of the 24 color chart patches, $C \in \mathbb{R}^{24\times3}$, from each image. Then, we compute an accurate color transformation, $T_C \in \mathbb{R}^{3\times3}$, between each pair of images from the main camera $\left(I_{1i}, I_{1j}\right)$ based only on the color chart values from the two images as

$$T_C^{1i\to1j} = \left(I_{1i}^T \, I_{1i}\right)^{-1} \, I_{1i}^T \, I_{1j}, \qquad (8)$$

and similarly, for image pairs from the second camera $\left(I_{2i}, I_{2j}\right)$ as

$$T_C^{2i\to2j} = \left(I_{2i}^T \, I_{2i}\right)^{-1} \, I_{2i}^T \, I_{2j}. \qquad (9)$$

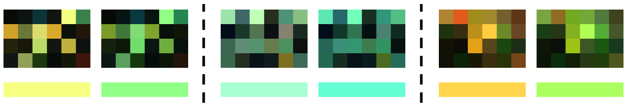


Figure 4: Three samples from our radiometric dataset. Each pair shows images from the main (left) and second (right) cameras. The ground truth illumination colors are presented in the bottom row.

Next, we use this bank of color transformations to augment our images by *re-illuminating* any given pair of images from the two cameras $\left(I_{1i}, I_{2i}\right)$ to match their colors to any target pair of images $\left(I_{1j}, I_{2j}\right)$, as follows:

$$I_{1i\to j} = I_{1i} \, T_C^{1i\to1j}, \qquad (10)$$

$$I_{2i\to j} = I_{2i} \, T_C^{2i\to2j}, \qquad (11)$$

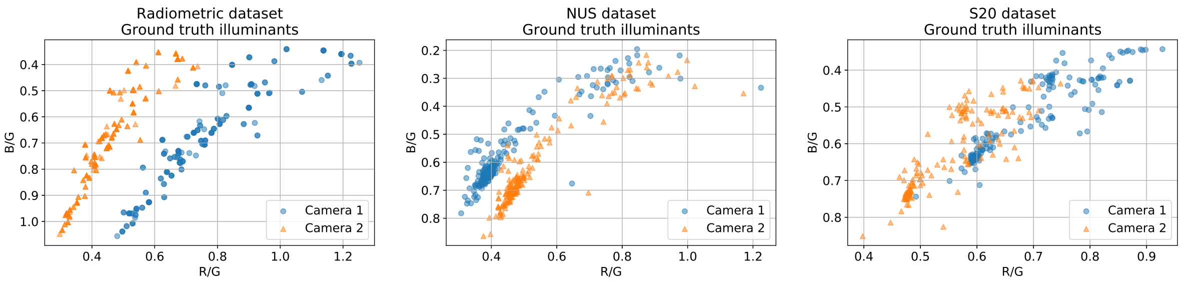
where $i \to j$ means re-illuminating image i to match the colors of image j. Using this illuminant augmentation method, we can increase the number of training image pairs from M to M^2. Fig. 3 illustrates an example of re-illuminating a pair of images given another target pair of images.

4. Experiments

To train our two-camera illuminant estimation network, we need a dataset of image pairs of the same scene captured with two different cameras under the same illumination. To our knowledge, there are no publicly available image datasets for color constancy captured using a two-camera system containing labelled ground truth illumination. To validate our method, we first present a synthetic radiometric dataset in Section 4.1. Next, in Section 4.2, we describe how to generate a quasi-real two-camera dataset from an existing single-camera color constancy dataset. Finally, we evaluate our method on a real two-camera image dataset that we captured using a Samsung S20 Ultra smartphone, in Section 4.3.

4.1. Radiometric dataset

To evaluate our method, we generate a synthetic dataset from radiometric data. According to the image formation model, the sensor response is the product of the scene illumination, the surface reflectance, and the sensor's spectral sensitivity, integrated over the visible spectrum. For data generation, we adopt the experimental procedure proposed in [6]. In particular, a scene illuminant and a random set of surface reflectances are selected from a hyperspectral dataset of lights and surfaces [7]. Two different camera sensors with different spectral sensitivity functions are chosen from the camera spectral sensitivity dataset of [33]. The RGB responses for both sensors can then be calculated by



(A) Radiometric dataset

(B) Quasi-real NUS dataset

(C) Our real dataset

Figure 5: Plots of ground truth illuminants for the two cameras for our (A) radiometric dataset, (B) quasi-real NUS dataset, and (C) real dataset.

simple numerical integration. The response induced by a pure reflector is treated as the corresponding ground truth. The advantage of this procedure is that it is easy to generate a large amount of labelled data to evaluate color constancy algorithms, and arrive at statistically meaningful performance measures.

The reflectance set of [7] consists of 1995 hyperspectral surface reflectance measurements of various natural objects, color charts, and so forth. The dataset of [7] also contains 87 different measured or synthesized illuminant spectra. The camera spectral sensitivity dataset of [33] contains the spectral sensitivity functions for 28 cameras, including mobile phone cameras. We select two sensors from this set to serve as our main camera and the second camera. To generate images, we choose a scene illumination and 24 different surfaces at random, and synthesize the raw-RGB sensor responses for both cameras. We generate thumbnail images of size 32 × 48 pixels. A few representative examples with associated ground truth are shown in Fig. 4. In total, we generate 18,000 pairs; 10,800 (60%) for training, and 3,600 (20%) each for validation and testing. A plot of the distribution of ground truth illuminants corresponding to the two cameras for 200 random samples is shown in Fig. 5(A). It is evident from the separation between the scatter points corresponding to the two cameras that the same illumination induces very different raw responses in the two sensors owing to the difference in their spectral sensitivity functions.

Method	Mean	Med	B25%	W25%	Q1	Q3
GW [15]	4.09	3.68	1.36	7.51	2.21	5.56
SoG [24]	4.56	4.11	1.51	8.41	2.43	6.21
GE-1 [46]	5.20	4.64	1.65	9.62	2.76	7.18
GE-2 [46]	5.41	4.69	1.72	10.25	2.83	7.37
WGE [29]	4.14	3.25	1.13	8.72	1.82	5.41
PCA [16]	4.55	3.09	1.03	10.67	1.68	5.85
WP [14]	5.49	4.96	1.83	10.02	2.94	7.48
Gamut Pixel [28]	3.68	3.07	1.05	7.30	1.70	5.10
Gamut Edge [28]	6.09	5.34	1.95	11.49	3.15	8.42
Ours (200 params)	2.80	2.20	0.72	5.87	1.19	3.81
Ours (470 params)	**2.65**	**2.00**	**0.64**	**5.72**	**1.07**	**3.61**

Table 1: Angular errors (degrees) on our radiometric dataset. B and W stand for best and worst, while Q1 and Q3 denote the first and third quantile, respectively. Best results are in bold.

For this experiment, we skip the alignment and downsampling steps of Section 3.1 since there is no misalignment, and compute our color transform for each pair from the 24 correspondences. We also omit the data augmentation procedure described in Section 3.4 since we have sufficient training examples. We use the Adam [37] optimizer with a learning rate of 10^{-4}. We train our network for 1 million epochs. The training process takes about 10 hours on a 32 GB nVidia Tesla V100 GPU. Table 1 reports statistics of the angular errors [20] obtained by our method, along with comparisons. The results of comparison methods were computed using open source codes downloaded from [1] or from the authors' webpages. Note that all comparison algorithms are single-image methods, and therefore were given the image from the main camera alone as input. For this experiment, we omit comparisons against deep learning methods since they are typically trained on natural images, whereas our images resemble color checker patches only. From Table 1, we can observe that our method performs better than well-established single-image illuminant estimation methods. Note that although we show illuminant estimation results only for the main camera for simplicity of comparison, our method, without loss of generality, can be used to predict the scene illuminant for the other camera. Please see the supplementary material for more details.

4.2. Quasi-real NUS dataset

In this section, we go a step further beyond synthetic data towards a more real dataset. In particular, we describe a procedure to generate a *quasi-real* two-camera dataset from an existing single-camera color constancy dataset. Towards this goal, we select the NUS [16] dataset, which has images

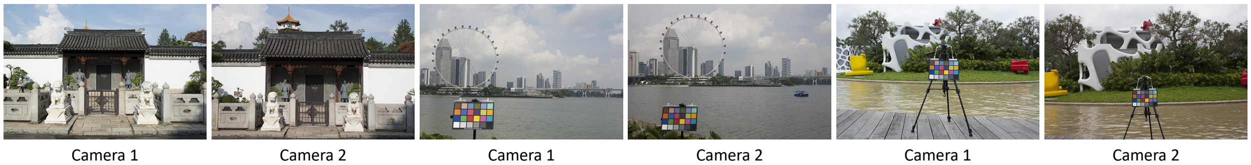


Figure 6: Sample matched pairs from the NUS dataset that we use to generate our quasi-real dataset.

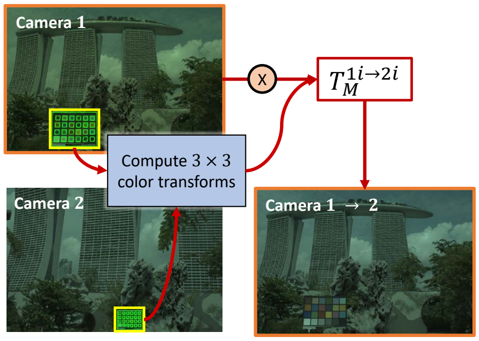


Figure 7: Our method for generating spatially-aligned two-camera image pairs from the NUS dataset. A color transform is used to map the image's colors from one camera to the other.

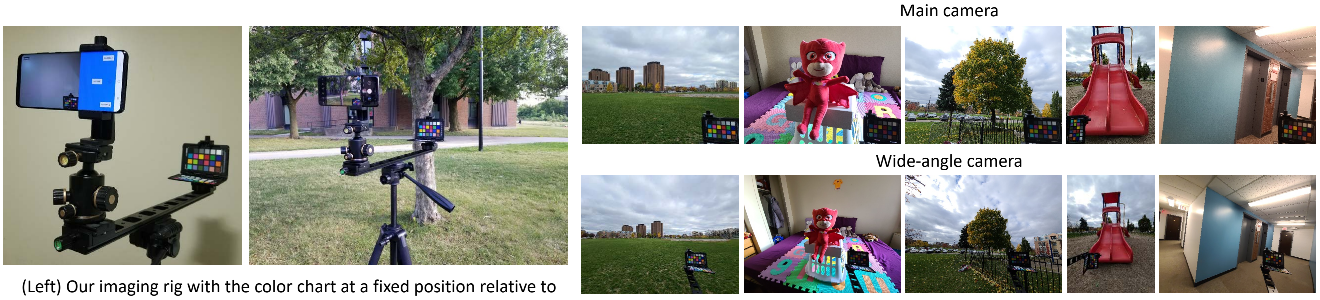
of the same scene mostly under the same illumination captured using different cameras. We choose the Nikon D5200 as the main camera while the Canon 1Ds Mark III serves as the second camera. We select only those images where the two cameras are observing the same scene with no visible changes in the illumination. After filtering, we obtain 195 matched pairs from the two cameras. All images in the NUS dataset have a Macbeth color chart placed in the scene. The ground truth scene illumination can be obtained from the achromatic patches in the color chart. A plot of the ground truth illuminants for the 195 images from the two cameras is shown in the plot of Fig. 5(B), and it can be observed that the two sensors record different measurements for the same illumination. A few representative examples of matched image pairs from the two cameras are shown in Fig. 6. Notice that some pairs have a significant change in viewpoint although the scene is the same. Therefore, we preprocess the data to generate our quasi-real dataset, as described next.

For each image pair (I_{1i}, I_{2i}) from the two cameras, we first compute an accurate 3×3 transform $T_M^{1i \to 2i}$ that maps the raw-RGB image from the main camera to the second camera using *only* the 24 correspondences from the color checker patches. Next, we apply this color transform on the main camera image to synthesize a new second camera image. This procedure is shown in Fig. 7. These two spatially aligned images constitute a pair in our quasi-real dataset.

We use a standard three-fold cross validation protocol to evaluate performance. For learning-based methods, including our own approach, we augment the training folds using the procedure described in Section 3.4. Testing is performed on the original unaugmented set. In particular, for each image pair, we generate another 99 randomly re-illuminated image pairs to obtain a total of 19500 pairs. The color chart is then masked out in all training, validation, and testing images. The results of our method, along with comparisons, are presented in Table 2. In addition to several classical methods, we also test against the recent learning approaches of [3, 10, 32, 9]. For all four learning methods, publicly available implementations provided by the authors were used to report results. The method of [3] is sensor-independent, and does not require re-training. The quasi unsupervised color constancy algorithm of [10], while inherently sensor-agnostic, can be fine-tuned if annotated training data is available. In Table 2, we report results both without and with fine-tuning, using the pre-trained models made available by the authors. For the fine-tuned result, we selected the appropriate pre-trained model for testing based on the three-fold partitioning indices of the NUS dataset used by the authors. For FC4 [32], we trained the model from

Method	Mean	Med	B25%	W25%	Q1	Q3
GW [15]	4.43	3.42	0.90	9.82	1.54	6.11
SoG [24]	3.31	2.63	0.70	7.20	1.18	4.17
GE-1 [46]	4.49	3.03	0.87	10.38	1.40	6.34
GE-2 [46]	4.99	3.28	0.94	11.83	1.54	6.65
WGE [29]	5.77	3.11	0.77	14.75	1.38	7.89
PCA [16]	4.01	2.68	0.69	9.20	1.22	6.07
WP [14]	4.49	3.47	0.93	9.99	1.42	6.09
Gamut Pixel [28]	5.99	3.70	0.90	14.95	1.41	8.65
Gamut Edge [28]	4.99	3.38	0.85	11.63	1.72	7.22
CM [18]	2.80	2.09	0.66	6.12	1.21	3.67
Homography [19] (SoG)	2.70	1.95	0.69	5.88	1.06	3.71
Homography [19] (PCA)	2.97	2.16	0.72	6.47	1.14	4.22
APAP [4] (GW)	2.64	2.00	0.60	5.99	1.02	3.26
APAP [4] (SoG)	2.49	1.75	0.60	5.61	0.88	3.14
APAP [4] (PCA)	2.77	1.83	0.60	6.45	0.94	3.49
SIIE [3]	2.04	1.55	0.51	4.41	0.80	2.80
Quasi U CC [10]	3.57	2.77	0.62	8.04	1.09	5.06
Quasi U CC finetuned [10]	2.68	1.72	0.57	6.25	0.98	3.67
FC4 [32]	2.65	2.06	0.67	5.69	1.12	3.49
FFCC [9]	2.44	1.50	0.40	5.87	0.75	3.19
Ours (200 params)	2.39	1.44	0.46	5.95	0.81	2.81
Ours (470 params)	1.91	1.24	0.36	4.78	0.62	2.22
Ours (1460 params)	**1.69**	**1.09**	**0.37**	**4.02**	**0.59**	**2.02**

Table 2: Angular errors (degrees) on the main camera from our quasi-real NUS [16] dataset. Best results are in bold.



(Left) Our imaging rig with the color chart at a fixed position relative to the camera. (Right) An outdoor scene being imaged using our setup.

Sample image pairs from our real dataset

Figure 8: Our data capture setup and representative examples from our dataset. Note that while illuminant estimation is performed on the raw-RGB sensor image, we show here the corresponding sRGB images to aid visualization.

scratch using the hyperparameters recommended by the authors. For FFCC [9], the hyperparameters were carefully tuned to achieve the best performance. In the literature, FC4 and FFCC are currently the best-performing methods across all color constancy datasets, including NUS. It can be observed that our model with 1460 parameters outperforms both FC4 and FFCC, as well as other competitors.

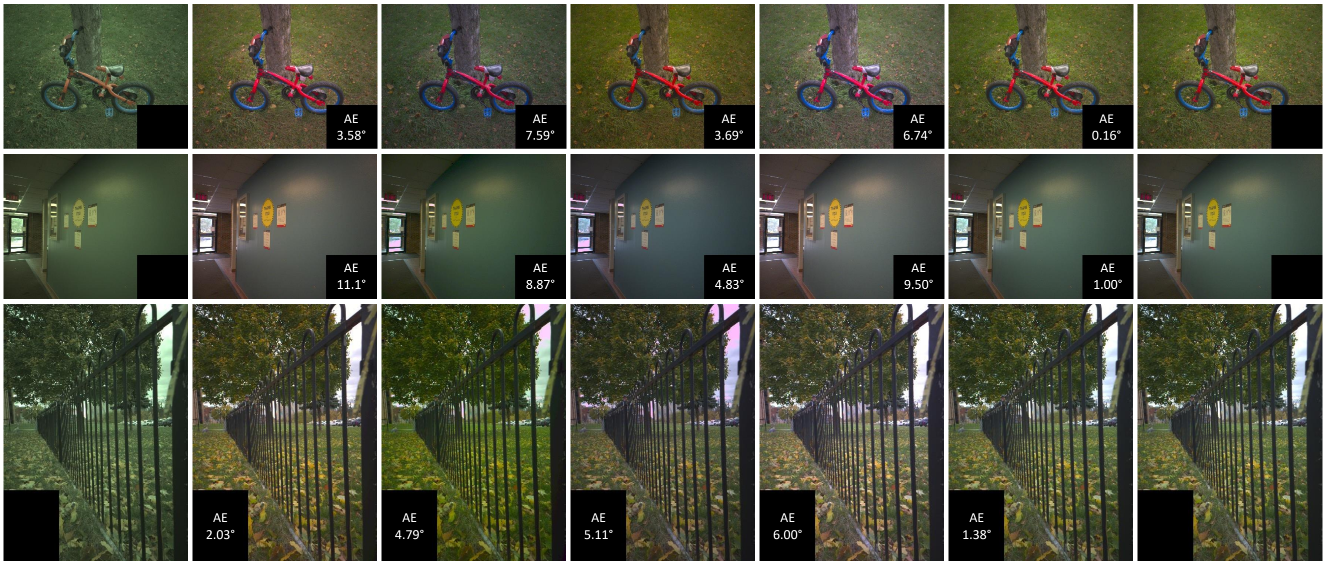
4.3. S20 real-image dataset

The final step in our evaluation is to collect and test on a real dataset of image pairs captured using a two-camera system. Towards this goal, we examined various recent smartphones with two rear-facing cameras. Our method requires access to the raw-RGB images from both cameras. The Samsung S20 Ultra is one smartphone we found that has the desired camera configuration and allows saving to the raw format. The S20 Ultra is equipped with a wide-angle rear-facing camera that provides a larger field of view than the main camera. The two camera sensors are different: the main camera is a Samsung HM1 sensor (108 MP, 3x3 Nonacell, $0.8\mu m$ pitch), while the second camera is a Samsung S5K2L3SX sensor (12 MP, $1.4\mu m$ pitch). While we do not have access to the sensors' CFA spectral sensitivities, it is easy to verify the CFAs are different by observing a color checker chart under the same controlled illumination and plotting the responses. See Fig. S1 of supplemental for more details on how we validate that the spectral sensitivities of the two cameras are different. We used image pairs from the main camera and the wide-angle camera for our experiments. We developed a simple Android application with the aid of the Camera2 API [30] to save the raw-DNG files from both cameras with a single button press. To obtain the ground truth, a Macbeth color chart was placed in every scene. For ease of ground truth labelling, we used a custom rig (see Fig. 8) that allows the color chart to be placed at a fixed position relative to the camera. This ensures that the color chart always occupies a fixed spatial location in the captured images. We collected a total of 156 image pairs, spanning a diverse range of lighting conditions and scene content. Some representative examples from our dataset are shown in Fig. 8. Fig. 5(C) shows a plot of the distribution of the ground truth illuminants for the two cameras. It is clear from the spread in the distribution that our working assumption of two-camera systems having different spectral profiles can likely hold true on real data.

Method	Mean	Med	B25%	W25%	Q1	Q3
GW [15]	3.25	2.55	0.90	6.94	1.46	3.73
SoG [24]	3.07	2.03	0.62	7.33	0.98	4.16
GE-1 [46]	4.79	3.91	0.94	10.72	1.49	6.35
GE-2 [46]	5.23	3.96	0.95	11.74	1.57	7.76
WGE [29]	6.17	4.76	0.85	14.17	1.50	9.79
PCA [16]	4.56	3.35	0.85	10.50	1.32	6.42
WP [14]	3.24	2.30	0.56	7.48	1.03	4.16
Gamut Pixel [28]	6.81	5.62	0.97	14.20	1.61	10.29
Gamut Edge [28]	5.00	3.60	0.94	11.20	1.46	6.58
CM [18]	3.51	2.64	0.66	7.64	1.25	4.83
Homography [19] (SoG)	3.43	2.38	0.46	7.93	1.13	4.90
Homography [19] (PCA)	4.35	3.12	0.58	10.09	1.05	6.35
APAP [4] (GW)	4.21	2.32	0.53	11.34	0.88	4.81
APAP [4] (SoG)	3.46	2.29	0.39	8.53	0.73	5.18
APAP [4] (PCA)	3.96	2.77	0.47	9.14	0.88	6.06
Linear regression	2.49	1.79	0.80	4.94	1.02	3.29
SIIE [3]	4.71	3.37	0.99	9.98	1.55	7.50
Quasi U CC [10]	3.94	2.66	0.71	9.16	1.21	5.71
Quasi U CC finetuned [10]	2.55	1.55	0.56	6.15	0.84	3.03
FC4 [32]	2.14	1.64	0.69	4.38	1.15	2.67
FFCC [9]	2.51	2.05	0.80	4.95	1.20	3.20
Ours (200 params)	1.73	1.29	0.37	3.75	0.70	2.32
Ours (470 params)	**0.94**	**0.69**	0.17	**2.14**	0.31	**1.24**
Ours (1460 params)	1.08	0.71	**0.16**	2.57	**0.27**	1.47

Table 3: Angular errors on the main camera from our S20 two-camera dataset. Best results are in bold.

As a preprocessing step, the raw-DNG images from our dataset were demosaiced and the black level was adjusted. The ground truth illumination was also extracted from the color chart. Since the field of view is different between the two cameras, before downsampling, we registered the images using a fixed pre-computed homography as described



(A) Raw-RGB input (B) SIIE [3] (C) Quasi U CC tuned [10] (D) FC4 [32] (E) FFCC [9] (F) Ours (G) Ground truth

Figure 9: Qualitative results from our real dataset. (A) Input raw-RGB image. (B-F) Results of [3, 10, 32, 9], and our method, respectively, after correcting the input images using the estimated illuminants. (G) Result of correcting using the ground truth illuminant. A gamma has been applied to the raw-RGB images in (A) for illustration. The results in the remaining columns have been rendered to the sRGB color space using [2] to aid visualization. Black boxes are used to mask out the color charts.

in Section 3.1. We then computed the transformation for each image pair. Data augmentation was performed as before to generate a total of 15600 image pairs.

The results on our real dataset are presented in Table 3. Three-fold cross validation was used as before, and training data was augmented for all learning-based methods. As a baseline comparison, we applied a linear regression model in place of our trained network. As seen from Table 3, linear regression yields good results, but our small network performs better because of non-linearities. Results for [3, 32, 9] were computed in the same manner as in Section 4.2. For [10], we fine-tuned the model for each fold using the parameters recommended by the authors. It can be observed that our model with just 200 parameters outperforms all competitors, including FC4 and FFCC. A few qualitative results, along with comparisons, are presented in Fig. 9. Table 4 reports results of training without and with our data augmentation technique. It is evident from the results that our augmentation framework improves performance.

Dataset	**Method**	Mean	Med	B25%	W25%
Real	Ours w/o	2.11	1.46	0.61	4.65
	Ours	**1.73**	**1.29**	**0.37**	**3.75**
NUS	Ours w/o	4.74	2.61	0.66	12.47
	Ours	**2.39**	**1.44**	**0.46**	**5.95**

Table 4: The results of our method with and without (w/o) data augmentation. All models shown have 200 parameters and were trained with a learning rate of 10^{-3}. Best results are in bold.

5. Conclusion

In this work, we take advantage of the availability of two rear-facing cameras, commonly used in modern smartphone design, to perform illumination estimation. Our approach leverages the differences in the sensor's spectral profile between these two cameras. In particular, we trained a lightweight neural network to estimate the scene illumination based on a 3×3 linear color transform that maps between the two cameras' colors. We demonstrated state-of-the-art illuminant estimation performance over contemporary single-image methods through extensive experiments on radiometric data, a quasi-real two-camera dataset generated from an existing single-camera dataset, and a real dataset that we captured using a two-camera smartphone. We believe our work may lead to design changes regarding how current camera devices perform illuminant estimation, leveraging the ubiquity of multi-camera devices. Our code, datasets involving radiometric, quasi-real, and real images from the S20 smartphone, and our trained models will be publicly released to the community. We hope our findings will spur further innovation in smartphone imaging through ideas that leverage multiple cameras.

References

[1] Color constancy : Research website on illuminant estimation. https://colorconstancy.com/source-code/index.html. Accessed: 2020-11-01. 5

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 8

[3] Mahmoud Afifi and Michael S. Brown. Sensor-independent illumination estimation for DNN models. In *BMVC*, 2019. 2, 6, 7, 8

[4] Mahmoud Afifi, Abhijith Punnappurath, Graham D. Finlayson, and Michael S. Brown. As-projective-as-possible bias correction for illumination estimation algorithms. *JOSA-A*, 36(1):71–78, 2019. 2, 6, 7

[5] Nikola Banic and Sven Loncaric. Color dog - Guiding the global illumination estimation to better accuracy. In *10th International Conference on Computer Vision Theory and Applications*, 2015. 2

[6] Kobus Barnard, Vlad Cardei, and Brian Funt. A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data. *TIP*, 11(9):972–984, 2002. 2, 4

[7] Kobus Barnard, Lindsay Martin, Brian Funt, and Adam Coath. A data set for color research. *Color Research & Application*, 27(3):147–151, 2002. 4, 5

[8] Jonathan T. Barron. Convolutional color constancy. In *ICCV*, 2015. 2

[9] Jonathan T. Barron and Yun-Ta Tsai. Fast fourier color constancy. In *CVPR*, 2017. 2, 6, 7, 8

[10] Simone Bianco and Claudio Cusano. Quasi-unsupervised color constancy. In *CVPR*, 2019. 2, 6, 7, 8

[11] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using CNNs. In *CVPR Workshops*, 2015. 2

[12] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Single and multiple illuminant estimation using convolutional neural networks. *TIP*, 26(9):4347–4362, 2017. 2

[13] David H. Brainard and William T. Freeman. Bayesian color constancy. *JOSA-A*, 14(7):1393–1411, 1997. 2

[14] David H. Brainard and Brian A. Wandell. Analysis of the retinex theory of color vision. *JOSA-A*, 3(10):1651–1661, 1986. 2, 5, 6, 7

[15] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1 – 26, 1980. 2, 5, 6, 7

[16] 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*, 31(5):1049–1058, 2014. 2, 5, 6, 7

[17] Graham Finlayson. Image recording apparatus employing a single CCD chip to record two digital optical images, 2006. US Patent 7,046,288. 2

[18] Graham D. Finlayson. Corrected-moment illuminant estimation. In *ICCV*, 2013. 2, 6, 7

[19] Graham D. Finlayson. Colour and illumination in computer vision. *Interface Focus*, 8, 2018. 2, 6, 7

[20] Graham D. Finlayson, Brian V. Funt, and Kobus Barnard. Color constancy under varying illumination. In *ICCV*, 1995. 5

[21] Graham D. Finlayson, Steven D. Hordley, and Peter Morovic. Chromagenic colour constancy. In *10th Congress of the International Colour Association*, 2005. 2, 3

[22] Graham D. Finlayson, Steven D. Hordley, and Peter Morovic. Colour constancy using the chromagenic constraint. In *CVPR*, 2005. 2, 3

[23] Graham D. Finlayson, Steven D. Hordley, and Ingeborg Tastl. Gamut constrained illuminant estimation. *IJCV*, 67(1):93–109, 2006. 2

[24] Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color Imaging Conference*, 2004. 2, 5, 6, 7

[25] David A. Forsyth. A novel algorithm for color constancy. *IJCV*, 5:5–35, 2004. 2

[26] Peter V. Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *CVPR*, 2008. 2

[27] Arjan Gijsenij and Theo Gevers. Color constancy using natural image statistics and scene semantics. *TPAMI*, 33(4):687–698, 2011. 2

[28] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Generalized gamut mapping using image derivative structures for color constancy. *IJCV*, 86:127–139, 2008. 2, 5, 6, 7

[29] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Improving color constancy by photometric edge weighting. *TPAMI*, 34(5):918–929, 2012. 5, 6, 7

[30] Google. Android Camera2 API. https://developer.android.com/reference/android/hardware/camera2/package-summary.html. Accessed: 2017-11-10. 7

[31] Daniel Hernandez-Juarez, Sarah Parisot, Benjamin Busam, Ales Leonardis, Gregory Slabaugh, and Steven McDonagh. A multi-hypothesis approach to color constancy. In *CVPR*, 2020. 2

[32] Yuanming Hu, Baoyuan Wang, and Stephen S Lin. FC4: Fully convolutional color constancy with confidence-weighted pooling. In *CVPR*, 2017. 2, 6, 7, 8

[33] Jun Jiang, Dengyu Liu, Jinwei Gu, and Sabine Süsstrunk. What is the space of spectral sensitivity functions for digital color cameras? In *WACV*, 2013. 4, 5

[34] Hamid Reza Vaezi Joze and Mark S. Drew. Exemplar-based color constancy and multiple illumination. *TPAMI*, 36(5):860–873, 2014. 2

[35] 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 Imaging Conference*, 2012. 2

[36] Hakki C. Karaimer and Michael S. Brown. A software platform for manipulating the camera imaging pipeline. In *ECCV*, 2016. 2

[37] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 5

[38] Zhongyu Lou, Theo Gevers, Ninghang Hu, and Marcel P. Lucassen. Color constancy by deep learning. In *BMVC*, 2015. 2

[39] Laurence T. Maloney. Physics-based approaches to modeling surface color perception. *Color vision: From genes to perception*, 1999. 1

[40] Simon Niklaus, Xuaner Cecilia Zhang, Jonathan T. Barron, Neal Wadhwa, Rahul Garg, Feng Liu, and Tianfan Xue. Learned dual-view reflection removal. *arXiv preprint arXiv:2010.00702*, 2020. 1

[41] Seoung Oh and Seon Kim. Approaching the computational color constancy as a classification problem through deep learning. *Pattern Recognition*, 2016. 2

[42] Dilip K. Prasad. Strategies for resolving camera metamers using 3+1 channel. In *CVPR Workshop*, 2016. 2

[43] Yanlin Qian, Ke Chen, Jarno Nikkanen, Joni-Kristian Kamarainen, and Jiri Matas. Recurrent color constancy. In *ICCV*, 2017. 2

[44] Charles Rosenberg, Alok Ladsariya, and Tom Minka. Bayesian color constancy with non-gaussian models. In *NeurIPS*. 2004. 2

[45] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *ECCV*, 2016. 2

[46] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *TIP*, 16(9), 2007. 2, 5, 6, 7

[47] Jin Xiao, Shuhang Gu, and Lei Zhang. Multi-domain learning for accurate and few-shot color constancy. In *CVPR*, 2020. 2

[48] Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Häne, Sean Fanello, and Rahul Garg. Du2net: Learning depth estimation from dual-cameras and dual-pixels. *arXiv preprint arXiv:2003.14299*, 2020. 1