

Physically-plausible illumination distribution estimation

Egor Ershov¹

Vasily Tesalin^{1,2,3}

Ivan Ermakov^{1,2}

Michael S. Brown⁴

¹IITP RAS

²MIPT

³Skoltech

⁴York University

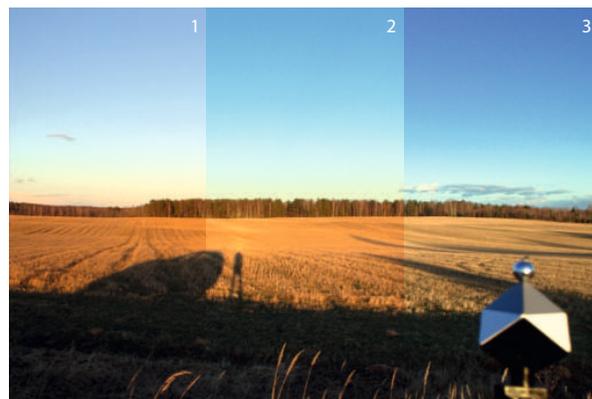
Abstract

A camera’s auto-white-balance (AWB) module operates under the assumption that there is a single dominant illumination in a captured scene. AWB methods estimate an image’s dominant illumination and use it as the target “white point” for correction. However, in natural scenes, there are often many light sources present. We performed a user study that revealed that non-dominant illuminations often produce visually pleasing white-balanced images and, in some cases, are even preferred over the dominant illumination. Motivated by this observation, we revisit AWB to predict a distribution of plausible illuminations for use in white balance. As part of this effort, we extend the *Cube++* illumination estimation dataset [12] to provide ground truth illumination distributions per image. Using this new ground truth data, we describe how to train a lightweight neural network method to predict the scene’s illumination distribution. We describe how our idea can be used with existing image formats by embedding the estimated distribution in the RAW image to enable users to generate visually plausible white-balance images.

1. Introduction

Illumination estimation (IE) is the core operation performed by a camera’s auto white balance (AWB) module. IE algorithms estimate the sensor’s response to scene illumination directly from an image in order to remove the color cast caused by the illumination. White balance correction is performed by scaling the image’s color channels such that the estimated illumination value becomes achromatic (i.e., $R=G=B$). Because the white balance correction procedure results in the estimated illumination being mapped to the achromatic color line, the estimated illuminant is often called the “white point” of the image. White balance correction simulates the mechanism of color constancy in the human visual system [17] to achieve realistic and aesthetically pleasing images.

IE algorithms operate under the assumption that a sin-



Correction using three different white points

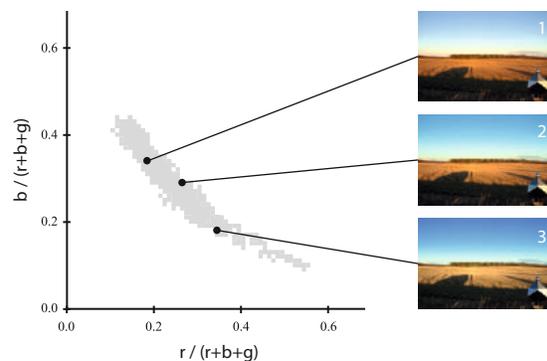


Figure 1. This figure shows the illumination distribution captured by the chrome ball on the SpyderCube calibration tool as described in Sec. 3. The distribution is plotted in the sensor’s chromaticity plane. Three different visually plausible white-balance corrections are shown based on illuminations (i.e., white points) sampled from the distribution.

gle light source serves as the dominant illumination in the scene. Most IE datasets for benchmarking and training have only a ground truth illumination label per image. This is achieved by placing an achromatic calibration object (e.g., a grey patch, grey ball, or grey cube) in the scene and measuring the average color recorded by the camera sensor of the calibration object.

However, most real scenes contain multiple light sources [1]. In outdoor environments, the sun, sky, and shadows can be treated as different illuminations. In indoor environments, artificial light sources (e.g., tungsten, fluorescent, LED) and natural light from the windows often illuminate the scene. In such cases, estimating a ground truth illumination value for a scene is highly dependent on the position and orientation of the gray patch. Moreover, white balance corrections performed using non-dominant environment illuminations often result in visually pleasing images (e.g., Fig. 1).

In this paper, we consider a new formulation of the IE problem that allows for a physically-plausible model of complex illumination in the scene. Instead of estimating a single dominant illumination, we propose to estimate a global *illumination distribution* for the whole scene. Sampling the illumination distribution allows us to generate a family of plausible white-balance corrections.

Contribution We leverage the existing Cube++ dataset [12] to establish ground truth illumination distributions for 4198 images by extracting color values from the chrome ball present in the Cube++ images. We perform a user study to verify that images white balanced with illuminations sampled from the extracted distributions are preferred over results obtained on samples from outside the distribution. Next, we examine how to an illumination distribution given a single input image. We describe how to adapt spatially-varying illuminant estimation algorithms and a DNN-based method to predict an illumination distribution. These baseline methods, however, produce suboptimal results. To address this, we propose a lightweight neural network trained on our dataset using a two-dimensional earth mover’s distance as a discrete version of the Wasserstein metric for the loss function. We show that a properly configured EMD provides compatibility with previous single-illuminant solutions estimated for angular and reproduction errors [16]. Our simple neural network provides better results than the baseline approaches. Finally, we discuss how an estimated illumination distribution could be embedded and used within the DNG RAW file format.

2. Related work

We describe prior work and datasets focused on single-illuminant and spatially-varying illumination estimation.

Single-illuminant estimation The vast majority of illumination estimation methods estimate a single illuminant per image. Given an estimated illumination represented as a sensor-specific color vector $[R, G, B]$, the white-balance operation is performed by dividing the image’s color values by the illumination. The problem can be reduced to predicting the 2D chromaticity representation of the illumination in the RAW-RGB space by predicting $(R/G, B/G)$.

Single illumination estimation is a long-standing problem in the computer vision field, with early methods addressing the problem using statistical features (e.g., [2, 20, 18, 15, 37, 21, 14]). Modern methods use neural network models which have proven to have state-of-the-art performance (e.g. [26, 3, 4, 6, 36]). Datasets used for single illumination estimation (e.g., Gehler-Shi [19], NUS [10], INTEL-TAU [29]) have a single ground truth (GT) illuminant associated per image. Deep-learning-based methods train on these datasets using either an $L2$ loss, reproduction [16] or angular error loss on the ground truth value.

The Cube++ dataset [12] is unique because it is labeled with two GT illumination values per image. The GT values are computed from the two sides of the SpyderCube calibration object placed in each scene, as shown in Fig. 1. Ershov et al. [12] showed that the angular distance between the two sides of the SpyderCube could be as high as 20 degrees, highlighting the weakness in the single illumination assumption.

Spatially-varying illumination estimation There are methods that seek to perform spatially-varying illumination correction. Such methods ideally estimate per-pixel illumination, which implicitly models the distribution of the scene’s illumination. A major challenge for spatially-varying methods is capturing annotated datasets for training and testing. As a result, many approaches rely on synthesized data by fusing images with known illuminations [22, 5] or using computer graphics [25].

Several works have captured datasets of real scenes with spatial GT labels. For example, work by Bleier et al. [7] sprayed entire scenes with achromatic paint to provide per-pixel illumination labels, but did so for only a handful of examples. Aghaei and Funt [1] placed a gray ball on a drone and flew it in a static scene capturing approximately 100 spatial samples of the environment’s illumination. Work by Kim et al. [27] prepared a large multi-illumination dataset by capturing several images of the same scene with a fixed camera with different light sources turned on or off. Work by Murmann et al. [33] used a similar strategy that captured scenes with diverse lighting conditions by adjusting a flash in different directions on a stationary camera. This latter work did not target computational color constancy but examined downstream tasks’ robustness to lighting variations.

While spatially-varying datasets provide dense GT values per image, no prior spatially-varying correction methods attempted to predict a illumination distribution for the purpose of global white-balance correction. In Sec. 7.2, we develop baseline methods fashioned after spatially-varying IE methods for comparison with our proposed neural network method.

3. Single illuminant vs. distribution

The assumption of a single illuminant model is an oversimplification, as most scenes contain multiple light sources. The image formation of a scene with n light sources can be expressed as:

$$\mathbf{r}(\mathbf{x}) = \int_{\omega} \sum_{i=1}^n \alpha_i(\mathbf{x}) S_i(\lambda) R(\lambda, \mathbf{x}) \chi(\lambda) d\lambda, \quad (1)$$

where \mathbf{r} is a sensor response, λ is a wavelength over the visible spectrum ω , $S_i(\lambda)$ is the spectra of i -th light source with weight $\alpha_i \geq 0$, $R(\lambda, \mathbf{x})$ is the scene reflectance at scene point \mathbf{x} , and $\chi(\lambda) = (\chi_R(\lambda), \chi_G(\lambda), \chi_B(\lambda))$ represents the sensor's channels spectral sensitivity.

Now let's consider the camera sensor's response to a diffuse achromatic object—such as a grey ball, where $R(\lambda, \mathbf{x}) = 1$ for every \mathbf{x} of the ball—illuminated by one, two, and n light sources as shown in Fig. 2.

Single light source S case: According to Eq. 1 an achromatic surface with only one light source can be modeled as:

$$\mathbf{r}(\mathbf{x}) = \alpha(\mathbf{x}) \int_{\omega} S(\lambda) \chi(\lambda) d\lambda,$$

where $\alpha(\mathbf{x})$ is a scale factor that depends on the surface geometry. An image histogram (for all points \mathbf{x}) plotted in the 2D-chromaticity space will result in a single point, which is the sensor's response to the light source S . The image's RGB histogram will consist of points lying on a ray starting from zero in the direction of light source S color.

Two illuminant S_1 and S_2 case: Two light sources give the following formation model:

$$\mathbf{r}(\mathbf{x}) = \int_{\omega} (\alpha_1(\mathbf{x}) S_1(\lambda) + \alpha_2(\mathbf{x}) S_2(\lambda)) \chi(\lambda) d\lambda,$$

where $\alpha_1(\mathbf{x}), \alpha_2(\mathbf{x}) \in R^+$ are weighting factors, which depend on light source intensity and geometry of the scene. For the two light source arrangement, the RGB histogram will have two beams starting from zero that coincide with light sources (S_1, S_2) colors and all their linear combinations. The histogram on the chromaticity plane will be a line segment.

Multiple S_i light source case: Given n light sources, S_i , we have:

$$\mathbf{r}(\mathbf{x}) = \int_{\omega} \sum_{i=1}^n \alpha_i(\mathbf{x}) S_i(\lambda) \chi(\lambda) d\lambda. \quad (2)$$

In this case, the sensor image histogram will be bounded in the chromaticity plane by the convex set defined by the S_i

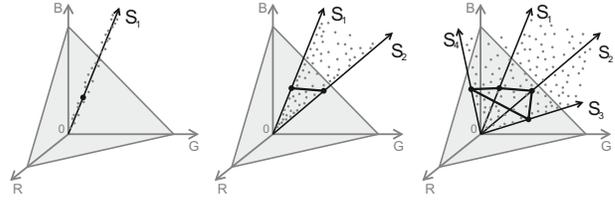


Figure 2. Illustration of RGB histograms of grey ball illuminated by a single light source (left), two light sources (center), and four light sources (right).

chromaticity values. Any value inside this set represents a mixture of the scene's light sources. The set of all possible mixtures defines the *illumination distribution*.

While the illumination distribution does not resolve if a particular light source (or combination) is dominant, it does indicate that white points falling within this distribution are plausible choices for white-balance correction. As we will show in Sec. 5, images corrected using values within the plausible illumination distribution are preferred to those outside the distribution.

The following section describes how we used the mirror ball in the Cube++ dataset to approximate the illumination distribution.

4. Illumination distribution dataset

As previously discussed, we extend the Cube++ dataset [12] consisting of 4890 RAW images captured with Canon 600D cameras. A SpyderCube calibration target was placed in each image. In the original Cube++ dataset, each image has been annotated with two global ground truth illumination values based on SpyderCube's top left and right surface appearance. Fig. 3 shows several example images from the dataset.

Fig. 3 also shows a zoomed region with the small chrome ball attached above the cube calibration target. We segment out the ball region using a manually marked-up polygon. A bounding box is assigned to this polygon to block out these pixels for training and testing. The average box size for the Cube++ dataset is 86×89 pixels.

Each mask is segmented into three regions. The first region is the lower half of the ball, which is discarded, as we assume the light sources are coming from above. Next, we segment overexposed or underexposed regions that provide unreliable sensor values. The remaining region (shown in green in Fig. 3) contains the pixel values that are used to estimate the illumination distribution. Images with masks where 50% of the pixels are over or underexposed are discarded. In the end, we had 4198 valid images. These were divided as follows: 2934 for training, 978 for validation, and 978 for testing.

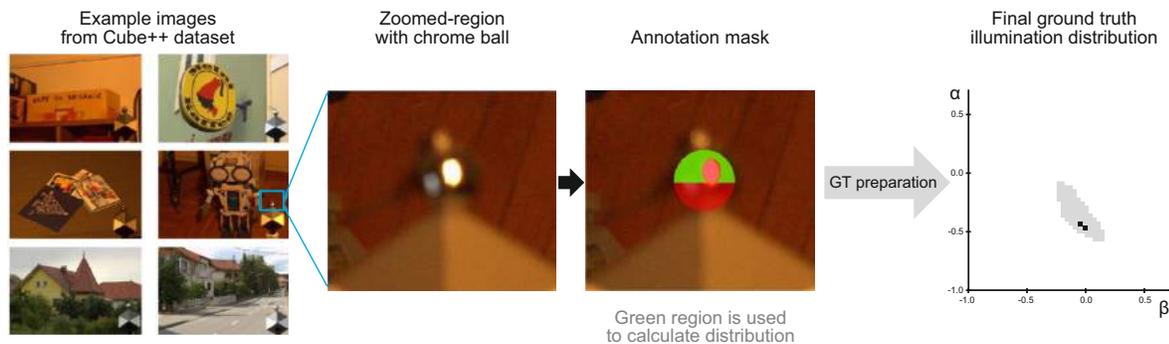


Figure 3. This figure shows examples from the Cube++ dataset. The chrome ball attached to the SpyderCube calibration target is shown. A mask is manually marked up for the chrome ball. The ground truth illumination distribution, represented as a 2D histogram, is estimated for each image using the RGB values that fall within the green mask region. Two black points on a chromaticity plane correspond to SpyderCube left and right grey sides.

The RGB-RAW values under the valid part of the mask are projected onto the chromaticity plane ($\alpha = (2B - (R + G))/(R + G + B)$, $\beta = (\sqrt{3}(R - G))/(R + G + B)$). To represent the illumination distribution, we generate a 2D histogram of the chrome ball pixels’ chromaticity values using a 116×100 grid, where the size of each grid cell equates to 0.03 in (α, β) coordinates. We pad both the x and y axes of the grid with zeros to make the final ground truth distribution per image 128×128 as shown in Figure 3. Because many images were captured in relatively low light conditions, we denoise the mirror ball pixels using BM3D [11] to improve the GT quality.

The extracted 2D histogram is binarized, so each non-zero histogram entry is treated as an equally probable illuminant. The binarized histogram is processed with simple morphology operations to remove outliers which is followed by small connected components filtration. Also, we add to the histogram points corresponding to SpyderCube grey edges. Finally, we use Graham’s method [24] to ensure the final distribution is a convex set, to be consistent with described physically-plausible illumination model (see Sec. 3). The GT preparation procedure is illustrated in Fig. 4.

5. Preference study

Using the GT illumination distributions described in the previous section, we want to determine if there are, in fact, multiple solutions from this distribution that may be preferred in addition to the dominant illumination. To answer this question, we performed a user study described in the following.

Study protocol We randomly sampled 100 images from our dataset. For each image, three illumination white-point chromaticities are sampled inside and three from outside the image’s GT illumination distribution. White-balance cor-

rection is applied to the image using these six selected white points.

A study was conducted by *pairwise comparisons*. The participants were presented with two images with different white-balance corrections and asked to choose the one they found more visually appealing. The controls available to the subjects are three buttons: “the left image is better”, “the right image is better” and “images are the same”. To deal with inattentive subjects, we include control tasks: pairs of identical images, for which it has to be selected that they are the same. Submissions containing wrong answers to the control tasks are discarded. For each scene each, two different corrections form a pair, and each pair is assessed by 10 people. *Scores* are calculated using the Borda Count method [35], which gives the share of pairs won by the correction.

Results The results of the user study are shown in Fig. 5. For images corrected using illuminations sampled from the ground truth illumination distribution, the mean opinion scores are generally not less than 0.6, while opinion scores of those corrected using white point outside of the distribution are distributed approximately uniformly between $[0, 0.6]$. This result reveals a strong negative correlation between the scores and the distance from the distribution. Images corrected using white points sampled from within the illumination distribution are preferred to the outside. Moreover, as we see a significant decline in scores at the border of the distribution, we can conclude that the quality of the ground truth illumination distributions in the dataset is reasonable.

To confirm the reliability of the study, we perform it again several months later. The scatter plot of scores from two runs in Fig. 6 suggests that the results are stable and reliable.

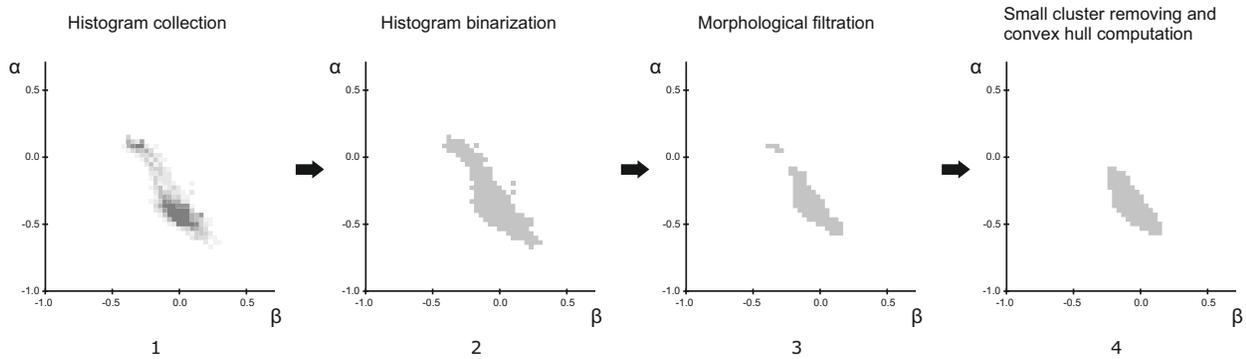


Figure 4. Example of the GT stages. (1) we first calculate the chromaticity histogram of the pixels within the chrome ball; (2) next, we perform histogram binarization; (3) Morphological filtering is used to remove small outliers. (4) The convex hull is computed to obtain the final distribution that will serve as the ground truth for the image.

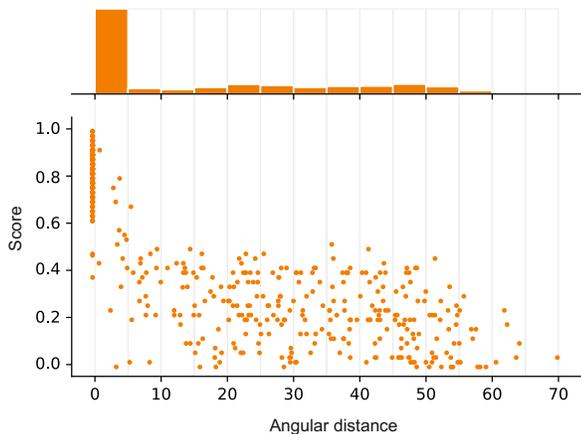


Figure 5. Scatter plot of the opinion study results. Each point corresponds to one correction. The angular distance for each correction is measured between the white point and the closest point within the illumination distribution (ID). For points within the ID, the angular distance equals zero.

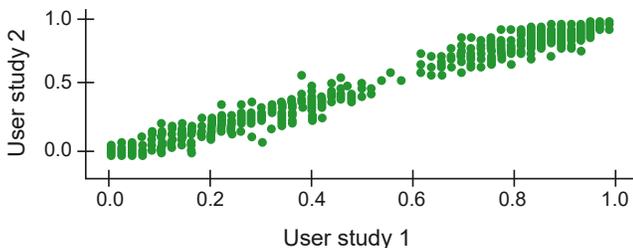


Figure 6. Comparison between two runs of the user study.

6. Evaluation metrics

Because we are working with distributions instead of a single value, we need to select a metric for training and evaluation on distributions. The most widely used met-

rics for distribution comparison are Kullback-Leibler divergence [28] and earth mover’s distance, also called Wasserstein or Monge–Kantorovich distance [34].

We opted to use the earth mover’s distance (EMD) applied to the 2D chromaticity histogram. For further details on this metric, see [31, 34]. Since the precise computation of EMD, even for the 2D case, is a time-consuming procedure [30], we have used an approximation of this metric based on fast Hough transform (FHT) [13]. The method works by computing 1D Earth mover’s distance between cumulative parallel projections of the histogram (calculated via FHT), and then averaging the results by angles, following the approach of [8]. Implementations are available with the code provided on our project page.

7. Methods for ID estimation

This section first describes several baseline approaches for illumination distribution (ID) estimation: single-illumination, spatially-varying, and a DNN-based method modified to predict a distribution. This is followed by a description of our neural network estimator.

7.1. Single illuminant estimation

Single illumination estimation methods may be evaluated using the collected ID dataset since a single estimate can also be treated as a simple distribution. We have implemented the following well-known statistical-based methods: GreyWorld [9], GreyEdge [37], MaxRGB [2]. We also include the FC4[26] deep-learning method. For the FC4 method, we fine-tuned the network on the Cube++ dataset using the left SynderCube value as ground truth.

7.2. Spatially-varying methods

As discussed in Sec. 2, spatially-varying methods estimate multiple illumination values over the image. We follow the approach proposed by Gisenji et al. [23], which

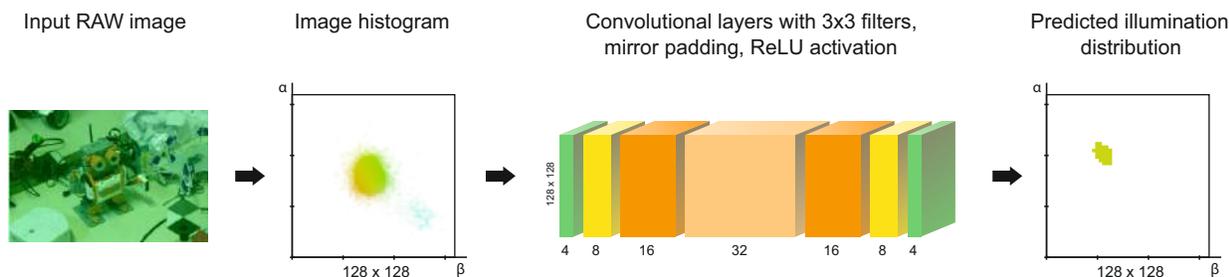


Figure 7. Proposed lightweight DNN for illumination distribution prediction. Input: image chromaticity histogram of 128×128 size. Output: illumination distribution of the same size.

uses traditional illumination estimation methods—namely, MaxRGB, Grey World, GreyEdge—to predict illumination within local regions. We have adapted their approach as follows:

1. Patch sampling *Different sampling strategies can be used, such as dense sliding window (SW), keypoint-based (KP) and segmentation-based. We utilize SW, where the image is split into equal-sized patches and KP sampling.*
2. Patch-wise illuminant estimation *Estimation is performed on each image patch to obtain a result using the methods described above.*
3. Combination of estimates *The patch estimations are aggregated to form the estimated illumination distribution.*

For the SW approach, we downsample the input image to resolution 128×192 and split it into 24 non-overlapping windows of size 32×32 .

For the KP approach, key point are detected using a Harris-Laplace detector [32] with the following parameters: the number of octaves in the scale-space pyramid is 8; the threshold for the Harris ‘cornerness’ measure is 0.01; the threshold for the Difference-of-Gaussians scale selection is 0.01; the maximum number of corners to consider is 500; the number of intermediate scales per octave is 2.

7.3. Modified FC4 neural network

Hu et al. [26] proposed a fully convolutional neural network (FC4) that predicted a spatial map of illuminations over the image and a corresponding confidence map. A confidence pooling mechanism aggregates the spatial map to produce a single illumination value. This approach is straightforward to generalize to produce an illumination distribution by applying a threshold to the confidence values and extracting the corresponding RGB values.

We used the pre-trained network¹ with SqueezeNet as a backbone. This was fine-tuned using the Cube++ dataset, using single illumination ground truth and angular error as the loss function. We fine-tuned the network for 100 epochs with batches of 8 images and used Adam optimizer with a learning rate equal to 0.0003. For tuning the hyperparameters (i.e., determining the confidence pooling threshold and histogram binarization threshold), a grid search was conducted. The best confidence map threshold was estimated as 0.3, and the optimal histogram threshold was 0.37.

7.4. Our neural network

We also designed a lightweight fully convolutional neural network that accepts as input an image histogram instead of an image. Our network architecture is illustrated in Fig. 7 and consists of eight convolution layers with ReLU activation functions. ReLU after the last layer imposes the non-negativity constraints for the output.

One reason for choosing the histogram domain over the image domain is the nature of convolutional neural networks. Our goal was to create a shallow network, thus, its perceptive field will be limited by the number of layers and their kernel sizes. However, if the network is to make the transition from the image domain to the histogram domain, every pixel of the input image should affect every pixel of the output distribution. If the input is a histogram, then this is not required. In addition, we observe that the illumination distribution histogram is often similar to the image histogram.

To train, we used the Adam optimizer with a learning rate 0.001 and weight decay 0.0001. Batches consisted of 8 images and was trained for 100 epochs. Histogram binarization threshold (set to 0.2) was obtained through grid search using the validation images. We used the fast EMD approach for the loss function as discussed in Sec. 6.

¹The code for the network, as well as a pre-trained net was obtained from GitHub [repo](#).

8. Experimental results

Table 1 shows that the statistical-based single illumination estimation methods (GreyWorld, MaxRGB, GreyEdge) have decent performance when evaluated using EMD against the 2D GT illumination distribution. To help understand the EMD result, the first row shows ground truth illumination estimated from the left side of the SyperCube (SP). The single-illuminant version of the FC4 neural network, trained on the SyperCube left GT value, produces a better result in terms of EMD. It is interesting to note that its error is not significantly larger than the error between ground truth *distribution* and one point estimate of the left edge of the SyperCube.

Table 1. 2D EMD results for methods that predict only a single illumination. To help gauge the results, the EMD score of the SyperCube’s left GT value is also plotted.

Algorithm	Mean	Median	10% worst	1% worst
<i>SPCube left</i>	2.1	1.9	4.7	6.8
Grey World	3.2	2.7	7.4	10.6
Max RGB	4.9	3.5	13.4	18.3
Grey Edge	3.1	2.7	7.1	10.2
FC4	2.5	2.3	4.1	7.9

However, in Table 2, it can be seen that FC4—when used to predict histograms (we will refer to this case as the “modified FC4”) —does not perform significantly better than the MaxRGB with the sliding window method. Our proposed neural network that takes an image histogram as input outperforms all baseline methods by a notable margin.

Fig. 8 and Fig. 9 show subjective comparisons between the proposed and baseline solutions. Fig. 8 shows an example of an outdoor scene, while Fig. 9 shows an indoor scene. Both figures show the input image, the ground truth

Table 2. 2D EMD results for different methods predicting illumination distributions. Methods are shown the sliding-window approach (SW) with different estimators, the keypoint-based (KP) approach with different estimators, and the two neural networks (NN) approaches.

	Algorithm	Mean	Median	10% worst	1% worst
SW	Grey World	3.3	2.9	7.5	10.7
	Max RGB	3.1	2.9	6.7	9.2
	Grey Edge	3.3	2.9	7.3	10.9
KP	Grey World	3.6	3.2	7.9	11.4
	Max RGB	3.5	3.2	7.4	11.4
	Grey Edge	3.6	3.1	7.9	11.7
NN	Modified FC4	3.0	2.5	5.4	8.7
	Our NN	2.0	1.6	2.8	6.6

illumination distribution, and the distribution estimated by the different methods. The first row shows white-balanced results corrected using the single-illumination approaches. The second row shows white-balanced results sampled from a distribution produced by the sliding-window method that used MaxRGB as its estimator. This approach was one of the better baseline performers (on par with the modified FC4). The final row shows images corrected using illuminations sampled from our neural network predicted distribution. We can see that our results produce plausible white-balance corrections. It is interesting to note that corrections generated by the proposed solution produce results that vary from low to high color temperatures for the outdoor case. Similarly, the indoor case exhibits notable variation in the white balanced images, but all appear visually pleasing.

Ablation An alternative to training our neural network with an EMD loss would be replacing it with an L_2 loss computed between the predicted and ground truth distributions. We performed this experiment by retraining our neural network approach with an L_2 loss. Table 3 shows the results, where the metric for evaluation is still the EMD. Note that the first row is repeated from Table 2. We can see that our method performs notably worse when relying on L_2 instead of the fast EMD loss.

Table 3. This table shows the results when our proposed method is trained using an L_2 loss versus the EMD loss.

Loss type	Mean	Median	10% worst	1% worst
<i>EMD</i>	2.0	1.6	2.8	6.6
L_2	2.7	2.2	4.8	10.2

Illumination distribution in practice Storing the estimated distribution inside the image container, specifically within a DNG file, is necessary to allow post-capture correction. A compact way of representing the distribution is to compute its convex hull points. The average number of convex hull points required for the ground-truth distributions in the proposed dataset was 7.6, while the maximum number was 35. Storing such information would be trivial in terms of the additional overhead required to include this in a DNG file. We envision such information could easily be used by RAW image processing software such as Photoshop and RawTherapee. While such software already allows users to adjust an image’s white point by manually tuning the color temperature and tint via slider controls, the user must rely on visual feedback for guidance. Directly sampling the estimated illumination distribution can help constrain results to lie within a plausible set of solutions, reducing the time needed to find good results.

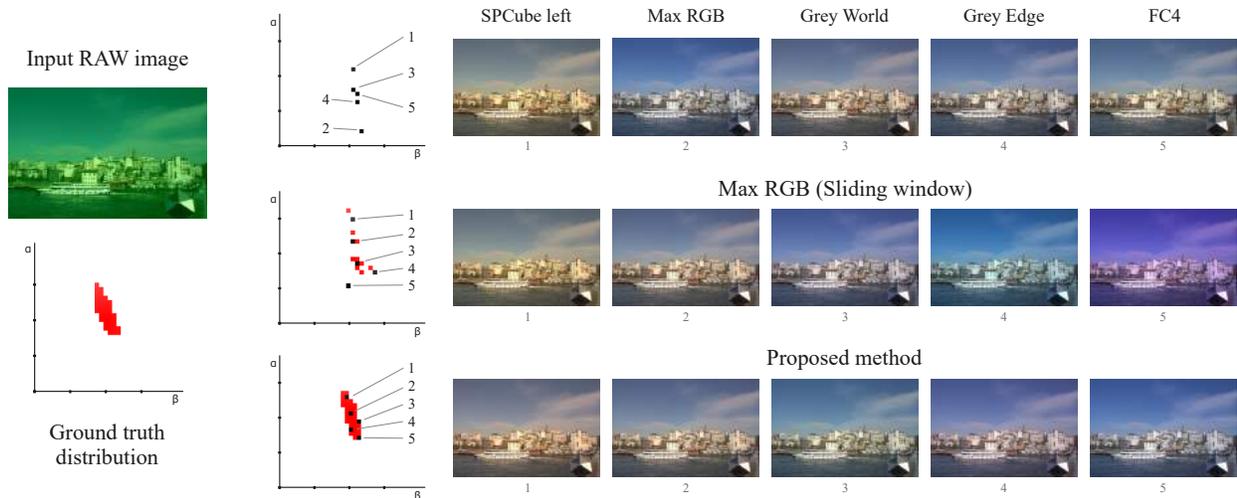


Figure 8. Different outdoor scene corrections: predicted illumination distribution using classical single illumination methods (first row), distribution estimated by sliding-window MaxRGB method (second row), and distribution estimated by the proposed lightweight DNN.

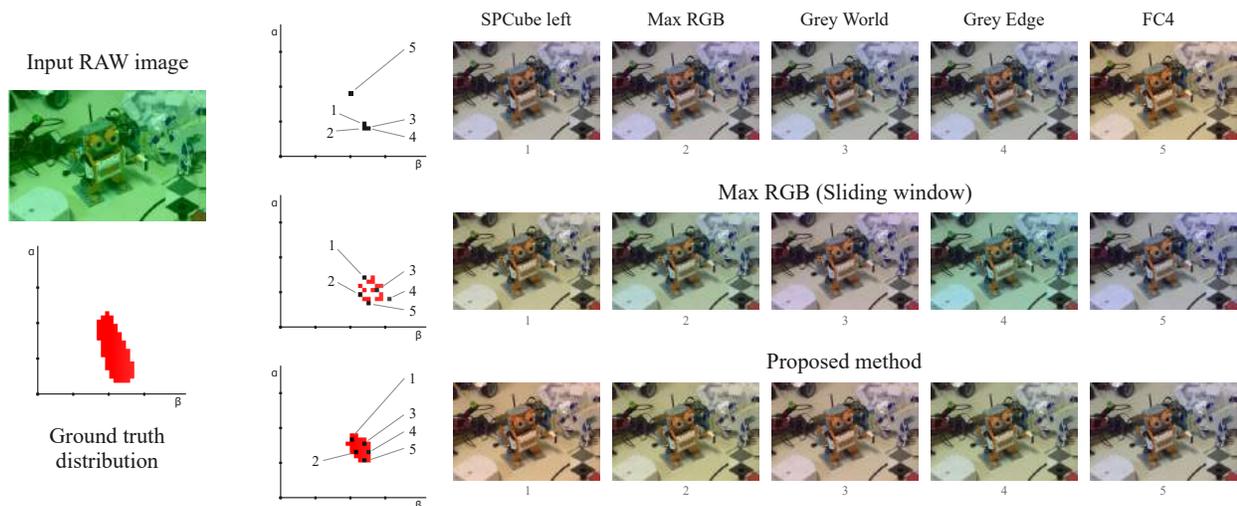


Figure 9. Different indoor scene corrections: predicted illumination distribution using classical single illumination methods (first row), distribution estimated by sliding-window MaxRGB method (second row), and distribution estimated by the proposed lightweight DNN.

9. Conclusion

We have introduced a new approach to address computational color constancy, which advocates estimating an illumination distribution versus a single illumination. Towards this goal, we have annotated 4198 images from the Cube++ dataset with ground truth illumination distributions. Using subjective studies, we justified that the collected dataset is suitable for correction using physically-plausible white points. In addition, we have described how to modify existing spatially-varying estimation methods to perform distribution estimation and introduced a lightweight neural network method that performs well on this task. Our dataset and code are available at: <https://github.com/createcolor/IDE>.

Acknowledgements Thanks to Irina Zhdanova for helping with the illustration preparation. We are grateful to Alex Savchik for fruitful conversations regarding network architectures.

References

- [1] Hoda Aghaei and Brian Funt. A flying gray ball multi-illuminant image dataset for color research. In *Color and Imaging Conference*, 2020. 2
- [2] Kobus Barnard, Vlad Cardei, and Brian Funt. A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data. *IEEE TIP*, 11(9):972–984, 2002. 2, 5
- [3] Jonathan T. Barron. Convolutional Color Constancy. In *ICCV*, 2015. 2

- [4] Jonathan T Barron and Yun-Ta Tsai. Fast Fourier Color Constancy. In *CVPR*, 2017. 2
- [5] Shida Beigpour, Christian Riess, Joost Van De Weijer, and Elli Angelopoulou. Multi-illuminant estimation with conditional random fields. *IEEE TIP*, 23(1):83–96, 2013. 2
- [6] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color Constancy Using CNNs. In *CVPR Workshops*, 2015. 2
- [7] Michael Bleier, Christian Riess, Shida Beigpour, Eva Eibenberger, Elli Angelopoulou, Tobias Tröger, and André Kaup. Color constancy and non-uniform illumination: Can existing algorithms work? In *ICCV Workshops*, 2011. 2
- [8] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015. 5
- [9] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of The Franklin Institute*, 310(1):1–26, 1980. 5
- [10] 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
- [11] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. BM3D Image Denoising with Shape-Adaptive Principal Component Analysis. In *Signal Processing with Adaptive Sparse Structured Representations*, 2009. 4
- [12] Egor Ershov, Alexey Savchik, Illya Semenov, Nikola Banić, Alexander Belokopytov, Daria Senshina, Karlo Košćević, Marko Subašić, and Sven Lončarić. The Cube++ Illumination Estimation Dataset. *IEEE Access*, 8:227511–227527, 2020. 1, 2, 3
- [13] Egor Ershov, Arseniy Terekhin, and Dmitry Nikolaev. Generalization of the fast hough transform for three-dimensional images. *Journal of Communications Technology and Electronics*, 63(6):626–636, 2018. 5
- [14] Graham D. Finlayson, Steven D. Hordley, and Paul M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *TPAMI*, 23(11):1209–1221, 2001. 2
- [15] Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, 2004. 2
- [16] Graham D. Finlayson and Roshanak Zakizadeh. Reproduction angular error: An improved performance metric for illuminant estimation. *Perception*, 310(1):1–26, 2014. 2
- [17] David H. Foster. Color constancy. *Vision Research*, 51(7):674–700, 2011. 1
- [18] Brian Funt and Lilong Shi. The rehabilitation of MaxRGB. In *Color and Imaging Conference*, 2010. 2
- [19] Peter V. Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *CVPR*, 2008. 2
- [20] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Computational color constancy: Survey and experiments. *IEEE TIP*, 20(9):2475–2489, 2011. 2
- [21] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Improving color constancy by photometric edge weighting. *TPAMI*, 34(5):918–929, 2012. 2
- [22] Arjan Gijsenij, Rui Lu, and Theo Gevers. Color constancy for multiple light sources. *TIP*, 21(2):697–707, 2011. 2
- [23] Arjan Gijsenij, Rui Lu, and T. Gevers. Color constancy for multiple light sources. *TIP*, 21:697–707, 08 2011. 5
- [24] Ronald L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1972. 4
- [25] Xiangpeng Hao, Brian Funt, and Hanxiao Jiang. Evaluating colour constancy on the new mist dataset of multi-illuminant scenes. In *Color and Imaging Conference*, 2019. 2
- [26] Yuanming Hu, Baoyuan Wang, and Stephen Lin. FC4: Fully convolutional color constancy with confidence-weighted pooling. In *CVPR*, 2017. 2, 5, 6
- [27] Dongyoung Kim, Jinwoo Kim, Seonghyeon Nam, Dongwoo Lee, Yeonkyung Lee, Nahyup Kang, Hyong-Euk Lee, ByungIn Yoo, Jae-Joon Han, and Seon Joo Kim. Large scale multi-illuminant (LSMI) dataset for developing white balance algorithm under mixed illumination. In *ICCV*, 2021. 2
- [28] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. 5
- [29] Firas Laakom, Jenni Raitoharju, Jarno Nikkanen, Alexandros Iosifidis, and Moncef Gabbouj. Intel-TAU: A color constancy dataset. *IEEE Access*, 9:39560–39567, 2021. 2
- [30] Haibin Ling and Kazunori Okada. An efficient earth mover’s distance algorithm for robust histogram comparison. *TPAMI*, 29(5):840–853, 2007. 5
- [31] Facundo Mémoli. The gromov–wasserstein distance: A brief overview. *Axioms*, 3(3):335–341, 2014. 5
- [32] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004. 6
- [33] Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. A dataset of multi-illumination images in the wild. In *ICCV*, 2019. 2
- [34] Yann Ollivier, Herve Pajot, and Cedric Villani. *Optimal Transportation: Theory and Applications*, pages 66–105. Number 413. 2014. 5
- [35] Nihar B. Shah and Martin J Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *The Journal of Machine Learning Research*, 18(1):7246–7283, 2017. 4
- [36] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep Specialized Network for Illuminant Estimation. In *ECCV*, 2016. 2
- [37] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *TIP*, 16(9):2207–2214, 2007. 2, 5