SENSOR DISTANCE LEARNING FOR CROSS-CAMERA COLOR CONSTANCY

Rafique Ahmed, Damien Muselet, Philippe Colantoni and Alain Trémeau

Université Jean Monnet Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France

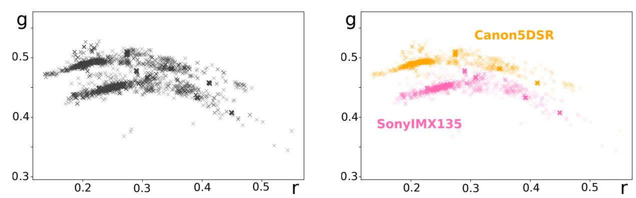
ABSTRACT

Computational color constancy has seen strong improvement these last years due to the emergence of large labeled datasets. However, the models trained on images acquired by some cameras show low generalization power when tested on images acquired by other cameras. Indeed, since the light chromaticities are device dependent, the training distribution is very spread out when mixing different sensors. In this paper, we propose to inform the network that this complex distribution is a set of simpler distributions, one for each considered camera. For this purpose, we create a Siamese architecture trained with a specific contrastive loss. This loss enforces the model to predict light chromaticities in the same sensor distribution, when considering images acquired by the same camera and in different distributions for images from different sensors. The key idea consists in learning a specific color distance that is sensitive to only sensor variations and not to lighting variations. This learned distance is a nice tool to control if two chromaticity points are in the same sensor distribution or not. We test this original training process in the context of cross-camera color constancy and we show that it outperforms the alternatives on three datasets.

Index Terms**—** Color constancy, Metric learning, Color distance, Sensor distribution, Color space

1. INTRODUCTION

The human visual system compensates for the effects of varying illumination and maintains constant color perception irrespective of lighting, leading to the phenomenon of color constancy. This is very convenient for humans because they can recognize objects and surfaces in scenes regardless of the current light spectral power distribution (SPD). This is very different when it comes to computer vision, when a camera is observing the world. The colors in the acquired images are highly sensitive to the lighting conditions and cannot be used as features to characterize surfaces. Computational color constancy consists in removing the impact of these varying lighting conditions on the acquired colors such that they appear as if the images have been acquired under a reference (typically white) light.



a) Sensor-blind *b) Sensor-aware*

Fig. 1. Ground-truth distribution for two cameras of the INTEL-TAU dataset [1].

Considering a Lambertian surface characterized by a spectral reflectance $\beta(\lambda)$, illuminated by a light source with a SPD $E(\lambda)$ and observed by a camera whose sensor sensitivities are denoted $R(\lambda)$, $G(\lambda)$ and $B(\lambda)$, the output C^k of the sensor $k \in \{R, G, B\}$ can be expressed as:

$$C^k = \alpha \int_\lambda E(\lambda)\beta(\lambda)k(\lambda)d\lambda, \quad (1)$$

where α is a constant across k depending on the light irradiance and the orientation of the surface.

Most of the computational color constancy algorithms split the task in two subtasks: first, estimating the light color (up to a multiplicative constant) whose components E^k are expressed as: $E^k = \int_\lambda E(\lambda)k(\lambda)d\lambda$, $\forall k \in \{R, G, B\}$, and second, removing the impact of this light on the colors of the pixels.

Note that during the first step, the light color is estimated in the color space of the current camera, so this estimation is device dependent. And when a model is trained with images from one camera, it provides poor results on images acquired by another camera, when no specific process is applied [2, 3, 4, 5]. This is because the spectral sensitivities of camera sensors are very diverse and provide very different color distributions. Fig. 1 shows the distribution of ground-truth light chromaticity in the ($r = \frac{E^R}{E^R+E^G+E^B}, g = \frac{E^G}{E^R+E^G+E^B}$) color space for two different cameras. In the right plot (b)), we can clearly see the discrepancy between the distributions of two different sensors. This explains why training on one device and testing on a second one can provide poor results.

This study was carried out as part of the PREMIERE project "Performing Arts in a new Area", https://premiere-project.eu/, funded by HORIZON-CL2-2021-HERITAGE-000201-04 (grant number 101061303-PREMIERE).

In order to increase the generalization power over cameras, the straightforward solution would consist in merging all the data from the available sensors and to train a model on this data without any sensor label, as shown in the left plot (a)) of Fig. 1 where all the points are gray. This is a difficult task because the merged distribution used for training is much more complex than each individual sensor distribution and we will show in the experiment that training from this "sensor-blind distribution" is not optimal. In this paper, we propose an original solution to train from the merged distribution while minimizing as much as possible the complexity of the task. Our main idea consists in informing the model that the complex training distribution is a mixture of simpler distributions (right plot b) of Fig. 1). By training on this "sensor-aware distribution", we help the model to understand that the complex task "estimating the light color from an image acquired by an unknown sensor" could be divided into two simpler sub-tasks: first, "estimating the current sensor distribution" and second, "predicting the light color within this distribution". We have the intuition that our specific training process helps the model to solve the problem with this strategy.

Nevertheless, since the characteristics of the used sensor are not known at test time (and the test sensor is different from the training sensors), we can not use any absolute label about each training sensor. Instead, we propose to use a Siamese architecture [6] that is fed with pairs of images and we force the predictions of two images acquired by the same sensor to be in the same distribution and images acquired by different sensors to be in different distributions. For this purpose, we design a specific color distance that is sensitive only to sensor variations and not to lighting variations. To be more specific, when evaluating the distance between two points in a chromaticity space, our learned distance considers only the direction corresponding to a sensor variation and neglects the direction of light source variations. These directions are learned on a specific dataset as detailed in Section 3. Our main contributions are twofold:

- We propose a Siamese deep architecture that solves the problem of sensor generalization for the cross-camera color constancy task.
- We design a new color metric in a chromaticity space that is sensitive to only sensor variation.

Our experimental results show that the proposed solution outperforms the current alternatives for color constancy across different sensors.

2. RELATED WORKS

Many works have been conducted to solve the color constancy task. A large amount of methods are based on empirical assumptions about statistics of real color images [7, 8, 9]. Other solutions are based on physical features and require to detect gray surfaces [10] or specularities [11]. The last group of algorithms resorts to machine learning models trained on real data, such as Gamut Mapping [12], patch-based approaches [13] or deep networks [14, 15]. Thanks to the numerous available annotated datasets, these learning-based solutions are currently the most effective. Nevertheless, as pointed out in the introduction they suffer from a lack of generalization across cameras.

Some solutions have been proposed to cope with this problem. Hernandez-Juarez et al. propose a Bayesian framework with a multi-hypothesis approach [16]. Their idea consists in selecting a set of candidate illuminants for a given image and to white-balance this image with each of these candidates. Then, the corrected images are provided as input to a network that independently predicts their probability to be correctly white-balanced. The predicted probabilities are used to estimate the correct light color. A multi-domain learning is proposed in [17] for color constancy over different devices. The deep architecture contains a camera-specific branch that projects the features extracted by the first module in a sensor-invariant space. The light color estimation is conducted in this common space. Likewise, Zhang et al. propose an adversarial training between a camera classifier and a feature extractor in order to encourage the emergence of device-invariant features for color constancy [18]. Xie et al. extract camera-independent scene statistics in order to match a test image with some training images [3]. Then color statistics are extracted from the white-balanced training image in order to indirectly estimate the current light color of the test image.

Afifi et al. also propose to improve the color constancy results on a new unseen camera and leverage a set of images from this camera to adapt a branch of their deep network [2]. Indeed, they exploit a similar approach as in [19] that learns filters and bias to detect a specific pattern in a $2D$ chromaticity histogram. These filters and bias are specific to a given camera and Afifi et al. show that they can obtain these filters from some images acquired by the test sensor. Convinced by the effectiveness of this recent solution, we build our approach on a similar architecture but, unlike this approach that requires a set of images to adapt the filters and bias for a new test, we propose a specific training process that allows to use only the test image at test time. For this aim, we resort to a Siamese architecture trained with a new color distance designed for this task.

3. OUR APPROACH

Current computational color constancy solutions are very effective when they are trained and tested on the same camera, because deep networks are able to learn the considered sensor distribution and to localize the current lighting conditions within this distribution. When training on several cameras and testing on a different one, the problem is much more complex because the training distribution is a mixture of sen-

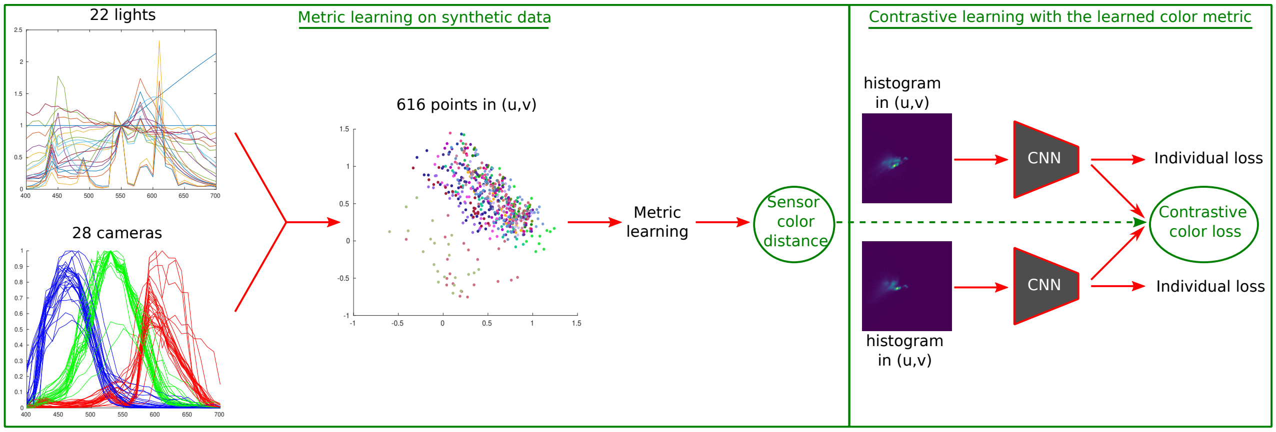


Fig. 2. The workflow of our approach. First, we learn a color metric that is sensitive to only sensor variations and second, we make use of this metric in a contrastive learning step.

sor distributions and the deep network does not know where to pick the current light color. Our idea consists in informing the model that several cameras have been used for training and enforcing it not to confuse between sensors in its predictions. For this aim, we transform the single branch architecture proposed in [2] into a Siamese architecture. This model is fed with pairs of histograms in order to control that two histograms coming from the same sensor provide outputs in the same sensor distribution, while two histograms coming from different sensors are associated with predictions lying in different sensor distributions. As illustrated in Fig. 2 and detailed below, two consecutive steps are required in our workflow: first, we learn a specific color metric that provides distances between sensor distributions, and second, we run a contrastive learning step where the loss is based on this sensor color distance.

3.1. Sensor color distance

Inspired by the approach in [2], we aim to localize the light chromaticity in the log-chrominance space (u, v), computed from the (r, g, b) chromaticity coordinates:

$$u = log(\frac{g}{r}), \quad v = log(\frac{g}{b}) \tag{2}$$

Since all the predictions are done in this space, we also learn our color metric in this space. Metric learning arises from the necessity to accurately compare examples [20]. Most of the existing work in metric learning is focused on learning a Mahalanobis-like distance between points $\mathbf{X}$ and $\mathbf{X}'$ of the form $d_M(\mathbf{X}, \mathbf{X}') = \sqrt{(\mathbf{X} - \mathbf{X}')^T M (\mathbf{X} - \mathbf{X}')}$ where M is a positive semi-definite (PSD) matrix to optimize.

In our case, we want to learn the best matrix M so that the associated distance is sensitive only to sensor variations and not to lighting variations. To be more specific, let consider a set of different chromaticity points in the (u, v) space. Each point $P_{i,m}$ is associated with the chromaticity of a white patch (flat spectral reflectance) illuminated with one specific light L_i and acquired with one specific camera C_m. For example, $P_{i,m}$ and $P_{i,n}$, with $n \neq m$, have been acquired by two different cameras and under the same illumination, while $P_{i,m}$ and $P_{j,m}$, with $i \neq j$ have been acquired by the same camera and under different illuminations.

Our aim is to learn a distance $d_M()$ in the (u, v) space such that $d_M(P_{i,m}, P_{j,n}) = \Delta_{m,n}$, $\forall\{i, j, m, n\}$, where $\Delta_{m,n}$ is a reference distance between the two cameras C_m and C_n, which does not depend on the lighting conditions L_i and L_j. In this paper, we define the reference distance between any two cameras as the Euclidean distance between their chromaticity outputs when they observe a gray patch (flat spectral reflectance) under a white light (flat spectral power distribution). Hence, whatever the lighting conditions, our trained distance $d_M()$ should return this reference value. Such a distance becomes very useful when it is necessary to evaluate the "sensor distance" between two predictions, independently of lighting conditions, as required in our pipeline.

Learning a global M matrix does not allow to take into account the structure of the chromaticity space and the non-linearity of each sensor distribution. Consequently, we propose to apply a non-linear polynomial expansion [21] of the chromaticity coordinates (u, v) before learning our metric. The order of this polynomial expansion is fixed to 3 after a $3-$fold cross-validation process. So, the $9D$ input coordinates of our metric learning step are: u, v, u^2, v^2, $u \times v$, u^3, v^3, $u \times v^2$ and $u^2 \times v$. Hence, the learned matrix M is 9×9.

To train our color metric, we exploit the dataset from Jiang et al. [22] providing the spectral sensitivities of 28 different cameras and we collected the spectral power distribution of 22 different common illuminants, among which $D65$, series of fluorescent lights, color leds, xenon flash, ... From this data, we have evaluated the 784 (28×28) reference distances $\Delta_{m,n}$ between all the pairs of available cameras (as defined above) and $N_c = 616$ (28×22) chromaticity points corresponding to the observation of a gray patch with all the camera/light pairs. Using this synthetic and diverse data allows us to learn a generic "sensor distance" that is not specific to the color constancy dataset under consideration.

Our metric learning process consists in randomly picking N_p pairs of coordinates among the N_c chromaticity points, in projecting the $2D$ (u,v) coordinates to $9D$ vectors $\mathbf{X}$ with the polynomial expansion and to minimize the loss function $\sum_1^{N_p}(\sqrt{(\mathbf{X}-\mathbf{X}')^T M(\mathbf{X}-\mathbf{X}')} - \Delta_{cam(\mathbf{X}),cam(\mathbf{X}')})^2$, where $cam(\mathbf{X})$ is the index of the camera used to construct the point $\mathbf{X}$. As explained in [23], this optimization problem has a closed form solution. To obtain a proper distance, the learned matrix M must be PSD and thus has to be projected onto the PSD cone after this optimization process. This is obtained by enforcing its eigenvalues to be non-negative.

After learning this matrix M on this synthetic data, we propose to use it in our loss function for the contrastive learning detailed below.

3.2. Contrastive learning

As illustrated in Fig. 2, the second step of our workflow consists in training a Siamese architecture by leveraging the learned color metric. Each branch of our Siamese network (see Fig. 3) is designed as the solution proposed in [2], for which the input is the $2D$ chromatic (u,v)-histogram of the considered image. This histogram feeds a Deep Neural Network (DNN) that outputs filters and bias that are convolved and summed to this histogram to get a $2D$ map. The light chromaticity estimation is the position associated with the global maximum in this map.

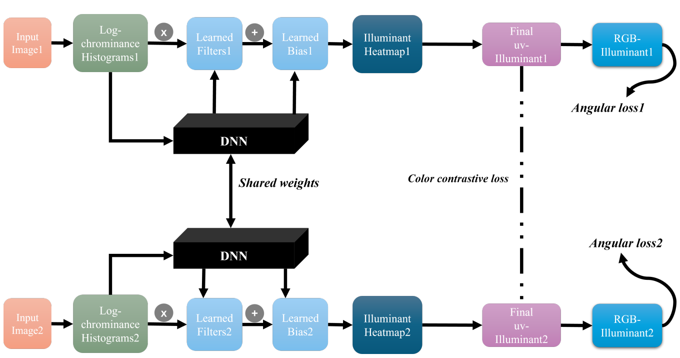


Fig. 3. Our Siamese architecture is trained with the supervision of each individual angular loss and our color contrastive loss. At test time, only one branch is preserved and fed with a single histogram.

In this paper, we transform this single branch model into a Siamese network where the two DNN share the same weights. During training, we feed our model with 2 histograms and we supervise the process with a color contrastive loss in addition to the classical individual angular losses. Each pair of histograms is associated with a binary label y that depends if they are characterizing images acquired by the same sensor ($y=1$) or by different sensors ($y=0$). When $y=1$ (same sensor), we want to enforce the model to predict two outputs $\hat{l}_{uv1}$ and $\hat{l}_{uv2}$ in the same sensor distribution, whereas when $y=0$ (different sensors), the model should predict two outputs in different distributions.

Since our learned metric ($d_M()$) has been designed to be sensitive to only sensor variations, it can measure the distance between sensor distributions. Consequently, we train our model by minimizing our learned sensor distance ($d_M(\hat{l}_{uv1},\hat{l}_{uv2})$) when $y=1$ (same sensor) and maximizing it when $y=0$ (different sensor). This is the aim of our color contrastive loss:

$$\begin{aligned}\mathcal{L}_c(\hat{l}_{uv1},\hat{l}_{uv2}) =& y * d_M(\hat{l}_{uv1},\hat{l}_{uv2})^2 + (1-y)\times \\ & max(T - d_M(\hat{l}_{uv1},\hat{l}_{uv2}),0)^2\end{aligned} \quad (3)$$

where T is a margin to avoid pushing the predictions to infinite distances [6].

Obviously, each branch is still supervised with an angular loss $\mathcal{L}_a$ between the predicted light chromaticity $\hat{l}_{uv}$ and the corresponding ground-truth to ensure that each prediction is accurate. The sum of this classical loss $\mathcal{L}_a$ with our color contrastive loss $\mathcal{L}_c$ aims to enforce the model to accurately predict the actual light chromaticity for each input histogram, but being more strict when the errors are in the direction that tend to confuse the sensor distributions. Thus, if the sensors used for training have large discrepancies between their distributions, the predicted distributions should not show small discrepancies between each other. This expected behavior of our contrastive loss will be checked in the experiments.

4. EXPERIMENTS AND RESULTS

4.1. Sensor metric learning

We propose to visualize our learned color metric $d_M()$ by drawing equidistant curves in the (r,g) space[1]. As usually done in color science, we have approximated these equidistant curves with ellipses. Each center of an ellipse is a reference color and all the points on the ellipse have the same $d_M()$ distance with this reference. It is worth mentioning that a classical Euclidean distance would have provided circles with constant radius as equidistant curves. Fig. 4 shows the ellipses obtained with our learned metric. We note that the ellipses are very different (elongations and orientations) between each other, showing that our sensor distance is able to adapt to the considered colors to be compared. Furthermore, we note that our ellipses clearly fit the distributions of Fig. 1 drawn in the same (r,g) space. Indeed, the ellipses show that our distance is very sensitive to the sensor variation direction, and much less sensitive to the lighting variation direction. Our distance completely fulfills its objective of being

[1] Our distance is learned in the (u,v) chromaticity space, but the transfer from one space to another is trivial.

sensitive only to sensor variations, regardless of illumination conditions. This is even more interesting, as our distance was not trained on the data in Fig. 1, but on generic synthetic data.

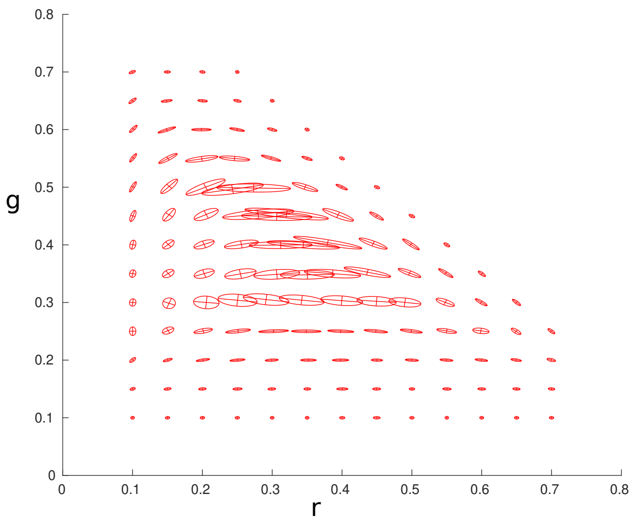


Fig. 4. Ellipses fitted to the equi-distant chromaticity points in the (r, g) space with our learned metric.

Our next experiments consist in training our Siamese network with this distance inserted in the contrastive loss of eq. 3.

4.2. Cross-dataset color constancy

We propose to test and compare the quality of our method on 3 different datasets widely used for the color constancy task: Gehler-Shi[24], INTEL-TAU[1] and NUS[8]. Gehler-Shi dataset contains 568 images captured with 2 cameras. INTEL-TAU dataset has $7,022$ images from 3 cameras and the NUS dataset comprises $1,736$ images from 8 cameras. These datasets contain raw RGB images and the corresponding ground truth for all images. For our problem, we propose to run 6 cross-dataset experiments, i.e. training on one dataset and testing on another one.

We use the same architecture as in [2] with a network depth of 2 and input histograms of size 128×128. We use Adam optimizer with batch size of 32 for 60 epochs with a learning rate of $5E-4$. The threshold in the contrastive loss is $T = 0.1$ and the global loss is a weighted sum of the two losses $\mathcal{L} = 0.5\mathcal{L}_a + 0.001\mathcal{L}_c$. At inference time, we pass only one sample to the model.

We compare our approach with the approach of [2] when feeding the network with a single histogram. We call this approach $C5_{1H}$ hereafter. We can see in Table 1 that our approach outperforms $C5_{1H}$ for all the experiments showing that our contrastive loss is appropriate for cross-camera color constancy.

Test →	G-S		NUS		I-T	
Train ↓	$C5_{1H}$	Our	$C5_{1H}$	Our	$C5_{1H}$	Our
G-S	-	-	3.18/2.53	2.80/2.22	3.12/2.24	2.93/2.09
NUS	3.40/2.61	3.34/2.59	-	-	3.75/2.94	3.18/2.47
I-T	3.33/2.22	2.82/1.95	3.22/2.48	2.98/2.41	-	-

Table 1. Angular errors (mean/median) for the cross-dataset experiments (G-S: Geher-Shi and I-T: INTEL-TAU).

As an ablation study, we replace our learned distance with a classical Euclidean distance in the contrastive loss of eq. 3. This is equivalent to forcing the predictions of the same sensor to be close in the chromaticity space, even in the case of large illumination changes between the two images. Table 2 shows that such a Siamese model does not help to improve the results of $C5_{1H}$ when using a simple Euclidean distance.

Test →	INTEL-TAU		
Train ↓	$C5_{1H}$	Euc. Dist.	Our
Gheler-Shi	3.12/2.24	3.14/2.27	2.93/2.09

Table 2. Angular errors (mean/median) on the Gehler-Shi dataset.

In order to check that our contrastive loss avoids overlapping the sensor distributions, we propose to display in Fig. 5 all the test predictions of the 2 cameras considered in Fig. 1 for the classical $C5_{1H}$ method and for our model. We clearly see that our contrastive loss forces the predictions of each sensor to be as far as possible from the other sensor prediction. This is even more interesting, as the sensor labels are not known at test time.

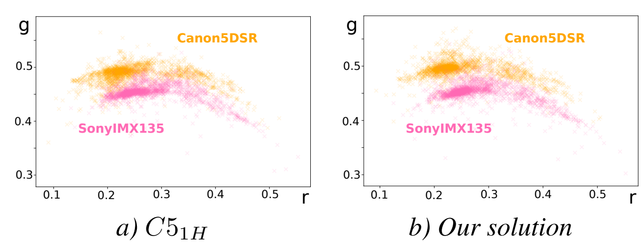


Fig. 5. Prediction distributions for the two cameras of Fig. 1

5. CONCLUSION

In this paper, we have proposed a specific architecture designed and trained in order to solve the problem of cross-camera constancy. Our original idea consists of informing the model that the complex training distribution is a mixture of simple distributions, one for each sensor. For this purpose, we have designed a Siamese architecture trained with a contrastive loss that is based on a specific sensor distance. The results show that this training process improves the single-branched alternative. Obviously, this sensor-aware training process can be applied to many other recent solutions.

6. REFERENCES

[1] F. Laakom, J. Raitoharju, J. Nikkanen, A. Iosifidis, and M. Gabbouj, "Intel-tau: A color constancy dataset," *IEEE access*, vol. 9, pp. 39560–39567, 2021.

[2] M. Afifi, J. T. Barron, C. LeGendre, Y.-T. Tsai, and F. Bleibel, "Cross-camera convolutional color constancy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1981–1990.

[3] M. Xie, P. Sun, Y. Lang, and M. Fang, "Camera-independent color constancy by scene semantics," *Pattern Recognition Letters*, vol. 171, pp. 106–115, 2023.

[4] S. Koskinen, D. Yang, and J.-K. Kämäräinen, "Cross-dataset color constancy revisited using sensor-to-sensor transfer," BMVC, 2020.

[5] Mahmoud Afifi and M. S. Brown, "Sensor-independent illumination estimation for dnn models," in *British Machine Vision Conference*, 2019.

[6] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 118–126.

[7] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, Sept 2007.

[8] D. Cheng, D. K. Prasad, and M. S. Brown, "Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution," *JOSA A*, vol. 31, no. 5, pp. 1049–1058, 2014.

[9] S. Gao, W. Han, K. Yang, C. Li, and Y. Li, "Efficient color constancy with local surface reflectance statistics," in *European Conference on Computer Vision (ECCV)*, 2014, vol. 8690, pp. 158–173.

[10] Y. Qian, J. Nikkanen, J. Kämäräinen, and J. Matas, "On finding gray pixels," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8054–8062, 2019.

[11] R. T. Tan, K. Nishino, and K. Ikeuchi, "Color constancy through inverse-intensity chromaticity space," *J. Opt. Soc. Am. A*, vol. 21, no. 3, pp. 321–334, Mar 2004.

[12] G.D. Finlayson, "Corrected-moment illuminant estimation," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2013, pp. 1904–1911.

[13] H.R.V. Joze and M.S. Drew, "Exemplar-based color constancy and multiple illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 860–873, May 2014.

[14] Y. Hu, B. Wang, and S. Lin, "Fc 4: Fully convolutional color constancy with confidence-weighted pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4085–4094.

[15] P. Das, Y. Liu, S. Karaoglu, and T. Gevers, "Generative models for multi-illumination color constancy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1194–1203.

[16] D. Hernandez-Juarez, S. Parisot, B. Busam, A. Leonardis, G. Slabaugh, and S. McDonagh, "A multi-hypothesis approach to color constancy," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2270–2280.

[17] J. Xiao, S. Gu, and L. Zhang, "Multi-domain learning for accurate and few-shot color constancy," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3258–3267.

[18] Z. Zhang, X. Kang, and A. Ming, "Domain adversarial learning for color constancy," in *IJCAI*, 2022, pp. 1693–1699.

[19] J. T. Barron, "Convolutional color constancy," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 379–387.

[20] Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng, "Distance metric learning with application to clustering with side-information," *Advances in neural information processing systems*, vol. 15, 2002.

[21] Graham D Finlayson, Michal Mackiewicz, and Anya Hurlbert, "Color correction using root-polynomial regression," *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1460–1470, 2015.

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

[23] L. Nicolle, J. Bonneton, H. Konik, D. Muselet, and L. Tougne, "Towards an electronic orientation table," in *International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications: VISAPP*, 2017, vol. 6, pp. 28–38.

[24] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp, "Bayesian color constancy revisited," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.