

# Truncated Edge-based Color Constancy

Simone Bianco

Department of Informatics, Systems and Communication  
University of Milano – Bicocca  
Milan, Italy

 orcid.org/0000-0002-7070-1545

Marco Buzzelli

Department of Informatics, Systems and Communication  
University of Milano – Bicocca  
Milan, Italy

 orcid.org/0000-0003-1138-3345

**Abstract**—In this paper we propose the truncated edge-based color constancy. It is based on, and extends, the edge-based framework by introducing the use of truncated Gaussian filters. The truncation level can be controlled with the use of a dedicated parameter that is added to the other three parameters existing in the edge-based framework, namely the derivative order, the standard deviation of the Gaussian filter, and the Minkowski norm. Experimental results on two standard dataset for color constancy show that the truncated edge-based framework allows to achieve the same or higher illuminant estimation accuracy of the edge-based framework considerably reducing the number of operations.

**Index Terms**—Color constancy, illuminant estimation, truncated Gaussian

## I. INTRODUCTION

The colors of the objects that we observe in a scene depend mainly on three different factors: i) the surface spectral reflectance of the objects; ii) the spectral power distribution of the illuminant; iii) the relative positions of the objects and the illuminant. The aim of color constancy, also referred to as illuminant estimation, is that of rendering the objects in the scene as if they were seen under a chosen neutral illuminant. It is therefore easy to imagine why this is a crucial step in digital camera pipelines and why many computer vision problems in both still images and videos make use of color constancy as a pre-processing step. Notwithstanding its simplicity, color constancy is a very challenging problem [1] since it is underdetermined. To solve this problem many different methods have been proposed in the state of the art, ranging from simple statistics-based approaches ([2]–[6]) to learning based approaches based on handcrafted features, and learning based approaches based on deep-learning exploiting both supervised ([7]–[9]) and unsupervised ([10], [11]) learning.

Although recent deep learning based approaches allow to estimate the illuminant with a high degree of accuracy, they have a large number of parameters to learn and therefore they need very large training sets. For this reason, simple statistics-based approaches are still used in practical applications, especially when inference speed is crucial.

In this work, starting from the edge-based framework for color constancy [4], we propose the truncated edge-based

framework that, by introducing the use of truncated Gaussian filters, allows to obtain illuminant estimates with the same or higher accuracy than those obtained by the edge-based framework, all the while considerably reducing the number of operations.

## II. PROPOSED METHOD

The proposed method starts from the edge-based framework introduced by van de Weijer et al. [4], whose general hypothesis is described as:

$$\left( \int \left| \frac{\partial^n f^\sigma(x)}{\partial x^n} \right|^p dx \right)^{1/p} = k e^{n \cdot p \cdot \sigma} \quad (1)$$

where  $n$  identifies the derivative order,  $\sigma$  is the standard deviation for a Gaussian filter  $G_\sigma$  that when applied with the convolution operator ( $*$ ) to the input image  $f(x)$  produces the filtered image  $f^\sigma(x) = G_\sigma * f(x)$ ;  $p$  is the order of the Minkowski norm, and  $f(x)$  is the input image. The above framework includes six color constancy algorithms that can be generated with different combinations of  $n$ ,  $p$ , and  $\sigma$ : Gray World (GW) [3] with  $[n, p, \sigma] = [0, 1, 0]$ , White Patch (WP) [2] with  $[n, p, \sigma] = [0, \infty, 0]$ , Shades of Gray (SoG) [12] with  $[n, p, \sigma] = [0, p, 0]$ , General Gray World (GGW) [4] with  $[n, p, \sigma] = [0, p, \sigma]$ , Gray Edge 1st order (GE1) [4] with  $[n, p, \sigma] = [1, p, \sigma]$ , and Gray Edge 2nd order (GE2) [4] with  $[n, p, \sigma] = [2, p, \sigma]$ . Among these six methods we can observe that only three make use of the Gaussian filter, i.e. GGW, GE1 and GE2.

In the edge-based framework the window size  $W_\sigma$  of the Gaussian filter depends from the chosen  $\sigma$ :

$$W_\sigma = (\lfloor B \cdot \sigma + 0.5 \rfloor \cdot 2) + 1 = (\lfloor 3\sigma + 0.5 \rfloor \cdot 2) + 1 \quad (2)$$

where  $B = 3$  is the default break-off sigma used. This means that the larger the  $\sigma$  of the filter, the larger will be the window size  $W_\sigma$ , and therefore more computations will be required.

In this work we want to remove the dependence of the window size from  $\sigma$ , making it explicitly parametrized by the truncation level  $t$ :

$$W_t = (t \cdot 2) + 1 \quad (3)$$

therefore potentially resulting in a truncated Gaussian. It is easy to see that the formulation in Equation 3 is equivalent to the original one in Equation 2 if we set  $t = \lfloor 3\sigma + 0.5 \rfloor$ .

 Source code available at: [https://github.com/simone-255-255-255/Truncated\\_EdgeBased\\_CC](https://github.com/simone-255-255-255/Truncated_EdgeBased_CC)

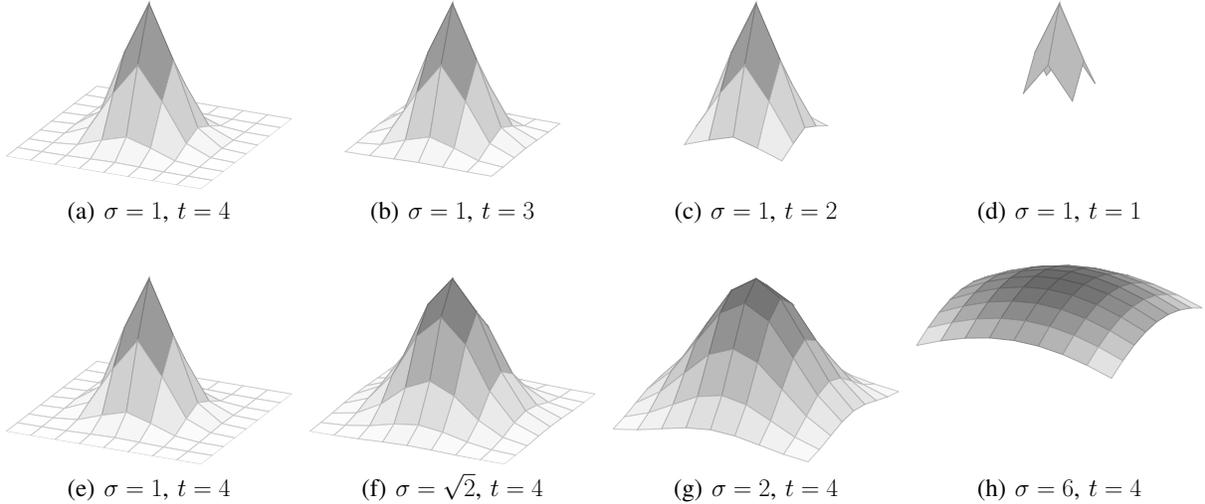


Fig. 1: Effect of different combinations of Gaussian standard deviation  $\sigma$  and truncation level  $t$ . Fixed  $\sigma$ , and varying  $t$  (first row):  $\sigma = 1, t = 4$  (a);  $\sigma = 1, t = 3$  (b);  $\sigma = 1, t = 2$  (c);  $\sigma = 1, t = 1$  (d). Varying  $\sigma$ , and fixed  $t$  (second row):  $\sigma = 1, t = 4$  (e);  $\sigma = \sqrt{2}, t = 4$  (f);  $\sigma = 2, t = 4$  (g);  $\sigma = 6, t = 4$  (h).

The advantage of the truncated formulation is that we can search the best parameter combination  $[n, p, \sigma]$  by explicitly controlling the window size of the filters involved, therefore controlling the number of operations required. Therefore we propose a truncated edge-based framework that extends the original edge-based framework [4] by exposing as fourth parameter the truncation level  $t$ . The general instance of an algorithm belonging to the truncated edge-based framework is therefore represented by  $[n, p, \sigma, t]$ . In other terms, by setting the truncation level  $t$  we are also setting the break-off sigma  $B$  in the solution interval of  $t = \lfloor B \cdot \sigma + 0.5 \rfloor$ , i.e.:

$$B \in \left[ \frac{t - 0.5}{\sigma}, \frac{t + 0.5}{\sigma} \right) \quad (4)$$

Examples of the effect of different combinations of  $\sigma$  and  $t$  are reported in Figure 1. From the plots reported it is possible to see that keeping  $\sigma$  fixed and decreasing  $t$  reduces the kernel size of the filter, while keeping  $t$  fixed and increasing  $\sigma$  permits to approximate the box filter (or average filter).

### III. EXPERIMENTS

#### A. Experimental setup

Experiments are performed on the ColorChecker dataset [13], composed of 568 images, and on the NUS dataset [14], composed of 1853 images. These are considered standard dataset in the specialized scientific community, and are characterized by a variety of scenes and illumination conditions [15]. All images include in the frame a 24-patch Macbeth Color Checker target that is masked for illuminant estimation. Multiple versions of the raw images and ground truth of the ColorChecker dataset have been proposed through the years. In this work we use the ‘recommended’ version by Hemrit et al. [16], [17].

#### B. Experimental results

In order to allow for a fine-grained sweeping of the parameters of each considered method, all the experiments are performed on a scaled version of the input images with the longest side equal to 200 pixels.

Parameter sweep is performed on the same grid for all the methods. The grid is created by the Cartesian product of  $\sigma = \{0.01, 0.1, 0.2, \dots, 1.0, 2.0, \dots, 50.0\}$  and  $n = \{1.0, 2.0, \dots, 10.0, \infty\}$  for a total of 660 (= 60·11) combinations. For the truncated versions a third parameter is swept, that represents the truncation level  $t = \{1, 2, \dots, 25\}$  for a total of 16500 (= 60·11·25) combinations.

In Table I we report the best results achieved by each method in terms of median and average recovery angular error. Next to each value it is also reported the parameter configuration to obtain it and the corresponding number of operations. Please notice that the best median and average values can be reached with different parameter settings. From the results relative to the ColorChecker dataset it is possible to notice that the truncated versions of the algorithms are able to achieve lower median errors than the original versions. Considering the average error, the difference between the two variants is very small, with the truncated version reaching always a value lower or equal to the corresponding original version. Results on the NUS dataset provide similar insights, with lower median errors for the truncated versions.

Figure 2 displays selected images from the ColorChecker dataset, under various types of correction. The first row presents an example from the Canon EOS-1DS camera, corrected with the GE1 and Truncated GE1 algorithm. The second row presents an example from the EOS 5D camera, corrected with the GGW and Truncated GGW algorithm. We select the best parametrization of the algorithms according to the median

TABLE I: Best median and average angular errors obtained by the considered methods on the ColorChecker dataset (top) and on the NUS dataset (bottom): for each entry we report the corresponding parameters and number of operations.

Dataset	Method	Median	Parameters	Operations	Average	Parameters	Operations
ColorChecker [13]	GGW	2.4958°	$[n, p, \sigma] = [0, 4, 0.1]$	1.13M	4.0184°	$[n, p, \sigma] = [0, 2, 0.1]$	1.13M
	Truncated GGW	2.4074°	$[n, p, \sigma, t] = [0, 8, 16, 1]$	1.13M	4.0184°	$[n, p, \sigma, t] = [0, 2, 0.1, 1]$	1.13M
	GE1	2.5958°	$[n, p, \sigma] = [1, 2, 0.9]$	2.54M	3.8785°	$[n, p, \sigma] = [1, 1, 3]$	6.59M
	Truncated GE1	2.4333°	$[n, p, \sigma, t] = [1, 3, 23, 3]$	2.73M	3.8524°	$[n, p, \sigma, t] = [1, 2, 27, 1]$	1.45M
	GE2	2.6935°	$[n, p, \sigma] = [2, 2, 9]$	27.10M	3.8895°	$[n, p, \sigma] = [2, 1, 5]$	15.52M
	Truncated GE2	2.5346°	$[n, p, \sigma, t] = [2, 4, 26, 6]$	6.83M	3.8874°	$[n, p, \sigma, t] = [2, 1, 5, 25]$	25.17M
Dataset	Method	Median	Parameters	Operations	Average	Parameters	Operations
NUS [14]	GGW	2.4268°	$[n, p, \sigma] = [0, \infty, 0, 6]$	1.64M	3.3088°	$[n, p, \sigma] = [0, 9, 0.1]$	1.13M
	Truncated GGW	2.3068°	$[n, p, \sigma, t] = [0, \infty, 23, 1]$	1.13M	3.3088°	$[n, p, \sigma, t] = [0, 9, 0.1, 1]$	1.13M
	GE1	2.1517°	$[n, p, \sigma] = [1, 2, 0.4]$	1.58M	3.1602°	$[n, p, \sigma] = [1, 2, 0.2]$	1.45M
	Truncated GE1	2.1455°	$[n, p, \sigma, t] = [1, 2, 0.9, 1]$	1.45M	3.1602°	$[n, p, \sigma, t] = [1, 2, 0.1, 1]$	1.45M
	GE2	2.1645°	$[n, p, \sigma] = [2, 2, 0.9]$	3.65M	3.1820°	$[n, p, \sigma] = [2, 2, 0.6]$	2.78M
	Truncated GE2	2.1581°	$[n, p, \sigma, t] = [2, 2, 5, 2]$	2.97M	3.1797°	$[n, p, \sigma, t] = [2, 2, 0.7, 1]$	2.01M



Fig. 2: Visualization of selected images corrected without and with truncated framework, compared to the version corrected with the ground truth, indicated in Angle-Retaining Chromaticity.

angular error reported in Table I. The reference ground truth for each image is also visualized, both in terms of corrected image and Angle-Retaining Chromaticity [18].

Another important aspect to be investigated is the best performance that can be achieved by the original and truncated algorithms when we impose an upper bound on the total number of operations. For each method, we compute the total number of operations required to process one input image of the ColorChecker dataset having the longest side equal to 200 pixels, with all the combinations of parameters above described. For each method we plot the minimum average and median angular error that can be reached by imposing an upper bound on the total number of operations, ranging from the lowest to the highest value required by each method in steps of 1000 operations. The corresponding plots in terms of millions of operations (MOps) are depicted in Figure 3.

From the plots reported in Figure 3 (a) it is possible to see that both the original and truncated GGW are able to achieve their best performance with the lowest number of operations (i.e. 1.13 MOps). Allowing them to use parameters

combinations that result in a higher number of operations does not improve the performance. We can observe how they obtain the same average error, while the truncated GGW obtains a lower median error with respect to the original GGW.

From the plots reported in Figure 3 (b) we can notice that with the lowest number of operations (i.e. 1.45 MOps) truncated GE1 is much better than the original GE1 both in terms of median and average error. While the number of operations required by both variants to achieve their best median error is quite similar (i.e. 2.73 MOps for truncated GE1 and 2.54 MOps for original GE1), the number of operations required for achieving their best average error is very different, i.e. 1.45 MOps for truncated GE1 and 6.59 MOps for original GE1. Furthermore we can observe that the ranges of performance obtainable by the two variants of GE1 are separated, with those of truncated GE1 being always lower than the corresponding ones of original GE1. In other words, it does not exist a number of operations that allows the original GE1 to equal the performance that can be obtained by truncated GE1.

From Figure 3 (b) it is possible to see that the performance

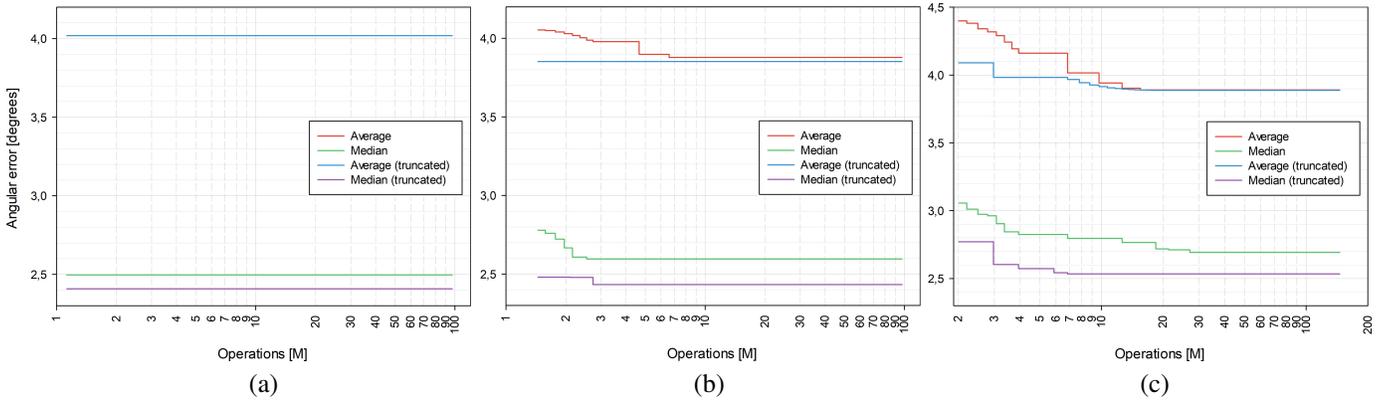


Fig. 3: Best average and mean recovery angular errors that can be obtained at different number of operations: (a) original GW vs truncated GGW; (b) original GE1 vs truncated GE1; (c) original GE2 vs truncated GE2.

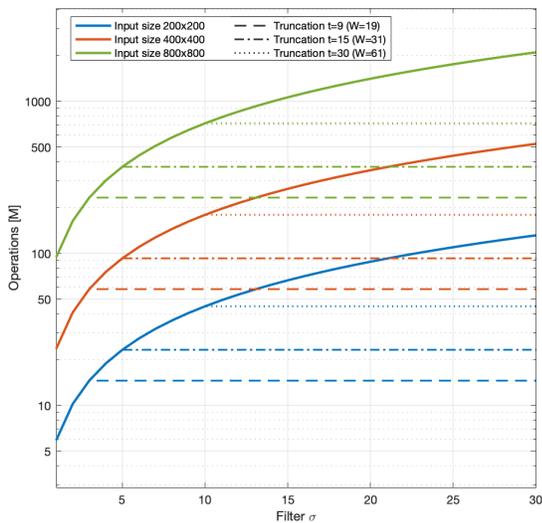


Fig. 4: Computational complexity (MOpS) of the GE2 algorithm by varying the Gaussian kernel standard deviation  $\sigma$ .

of the truncated GE2 are better than those of the original GE2 for all numbers of operations. While the average errors tends to be very similar when we allow a number of operations above 19 MOpS, the difference in median errors remains consistent regardless of the number of allowed operations. In particular, the original GE2 needs 12.62 MOpS to obtain a median error close to the one that the truncated GE2 is able to obtain with just 2.01 MOpS. Moreover the original GE2 is not able to obtain a median error close to what truncated GE2 obtains with 2.98 MOpS regardless of the considered number of operations. Concerning the average error instead, the original GE2 requires 6.84 MOpS to outperform what the truncated GE2 obtains with 2.01 MOpS, and 9.73 MOpS to outperform what the truncated GE2 obtains with 2.98 MOpS.

In Figure 4 we report the number of operations as a function of the kernel size. We analyzed GE2, which is the most expensive among the algorithms considered, by changing the input image size (i.e.  $200 \times 200$ ,  $400 \times 400$ ,  $800 \times 800$ ),

the filter  $\sigma$  in the range  $[1, 30]$  (corresponding to a filter size  $W_\sigma \in [7, 151]$ ). Furthermore we also report what happens to the number of operations at three different truncation levels  $t = \{9, 31, 61\}$ .

#### IV. CONCLUSIONS

We proposed the truncated edge-based color constancy, which is based on, and extends, the edge-based framework by introducing the use of truncated Gaussian filters. The truncation level is explicitly regulated with a dedicated parameter that specifies the filter window size, as such implicitly regulating the Gaussian break-off parameter. We conducted an extensive experimentation to investigate the range of performance achievable by the truncated edge-based framework in relation to the computational effort. We showed that thanks to this simple, yet effective, parametrization, it is possible to achieve equivalent or higher accuracy than the original edge-based framework, while reducing the required operations.

These results can be framed within a line of research in computational color constancy that aims at high-efficiency for integration in embedded devices. To this extent, in the future we will consider exploiting the truncated edge-based framework in the domain of video color constancy [19], [20].

#### REFERENCES

- [1] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer, "Computational color constancy: Survey and experiments," *IEEE transactions on image processing*, vol. 20, no. 9, pp. 2475–2489, 2011.
- [2] Edwin H Land and John J McCann, "Lightness and retinex theory," *Josa*, vol. 61, no. 1, pp. 1–11, 1971.
- [3] Gershon Buchsbaum, "A spatial processor model for object colour perception," *Journal of the Franklin institute*, vol. 310, no. 1, pp. 1–26, 1980.
- [4] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij, "Edge-based color constancy," *IEEE Transactions on image processing*, vol. 16, no. 9, pp. 2207–2214, 2007.
- [5] Arjan Gijsenij and Theo Gevers, "Color constancy using natural image statistics and scene semantics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 687–698, 2010.
- [6] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer, "Physics-based edge evaluation for improved color constancy," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 581–588.

- [7] Simone Bianco, Claudio Cusano, and Raimondo Schettini, "Color constancy using cnns," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 81–89.
- [8] Wu Shi, Chen Change Loy, and Xiaoou Tang, "Deep specialized network for illuminant estimation," in *European conference on computer vision*. Springer, 2016, pp. 371–387.
- [9] Yuanming Hu, Baoyuan Wang, and Stephen Lin, "Fc4: Fully convolutional color constancy with confidence-weighted pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4085–4094.
- [10] Marco Buzzelli, Joost van de Weijer, and Raimondo Schettini, "Learning illuminant estimation from object recognition," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3234–3238.
- [11] Simone Bianco and Claudio Cusano, "Quasi-supervised color constancy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12212–12221.
- [12] Graham D Finlayson and Elisabetta Trezzi, "Shades of gray and colour constancy," in *Color and Imaging Conference*. Society for Imaging Science and Technology, 2004, vol. 2004, pp. 37–41.
- [13] 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.
- [14] 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*, vol. 31, no. 5, pp. 1049–1058, 2014.
- [15] Marco Buzzelli, Simone Zini, Simone Bianco, Gianluigi Ciocca, Raimondo Schettini, and Mikhail K. Tchobanou, "Analysis of biases in automatic white balance," *Color Research & Application*, 2022.
- [16] Ghalia Hemrit, Graham D Finlayson, Arjan Gijsenij, Peter Gehler, Simone Bianco, Brian Funt, Mark Drew, and Lilong Shi, "Rehabilitating the colorchecker dataset for illuminant estimation," in *Color and Imaging Conference*. Society for Imaging Science and Technology, 2018, vol. 2018, pp. 350–353.
- [17] Ghalia Hemrit, Graham D Finlayson, Arjan Gijsenij, Peter Gehler, Simone Bianco, Mark S Drew, Brian Funt, and Lilong Shi, "Providing a single ground-truth for illuminant estimation for the colorchecker dataset," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1286–1287, 2019.
- [18] Marco Buzzelli, Simone Bianco, and Raimondo Schettini, "Arc: Angle-retaining chromaticity diagram for color constancy error analysis," *JOSA A*, vol. 37, no. 11, pp. 1721–1730, 2020.
- [19] Marco Buzzelli and Ilaria Erba, "On the evaluation of temporal and spatial stability of color constancy algorithms," *JOSA A*, vol. 38, no. 9, pp. 1349–1356, 2021.
- [20] Yanlin Qian, Jani Käpylä, Joni-Kristian Kämäräinen, Samu Koskinen, and Jiri Matas, "A benchmark for burst color constancy," in *European Conference on Computer Vision*. Springer, 2020, pp. 359–375.