Using the red chromaticity for illumination estimation

Nikola Banić and Sven Lončarić
Image Processing Group
Department of Electronic Systems and Information Processing
Faculty of Electrical Engineering and Computing
University of Zagreb, 10000 Zagreb, Croatia
E-mail: {nikola.banic, sven.loncaric}@fer.hr

***Abstract*—Achieving color invariance to illumination is known as color constancy and it is implemented in most digital cameras. There are statistics-based and learning-based computational color constancy methods and the latter ones are known to be more accurate. For a given image these methods extract some features and since for some methods calculating and processing these features can be computationally demanding, this renders such methods slow and very often impractical for hardware implementation. In this paper simple, yet very powerful features for color constancy based on the red chromaticity are presented. A new color constancy method is proposed and it is demonstrated how an existing one can be simplified. In both cases state-of-the-art results are achieved. The results are presented and discussed and the source code is available at http://www.fer.unizg.hr/ipg/resources/color_constancy/.**

***Keywords*—Chromaticity, color constancy, illumination estimation, image enhancement, white balance.**

I. Introduction

One of many abilities of the human visual system (HVS) is color constancy, which is used to recognize the object colors regardless of the scene illumination [1]. Achieving computational color constancy is important for making the object colors in the image invariant to illumination. Fig. 1 shows a possible difference in colors when the scene is lit by different illuminants. The most important step for a successful removal of illumination influence is an accurate illumination estimation. This is followed by chromatic adaptation, which uses this estimation to correct the image colors, which is relatively easy. For illumination estimation the following image f formation model with Lambertian assumption is often used:

$$f_c(\mathbf{x}) = \int_{\omega} I(\lambda, \mathbf{x}) R(\mathbf{x}, \lambda) \rho_c(\lambda) d\lambda \tag{1}$$

where c is a color channel, $\mathbf{x}$ is a given image pixel, λ is the wavelength of the light, ω is the visible spectrum, $I(\lambda, \mathbf{x})$ is the spectral distribution of the light source, $R(\mathbf{x}, \lambda)$ is the surface reflectance, and $\rho_c(\lambda)$ is the camera sensitivity of the c-th color channel. Assuming uniform illumination, $\mathbf{x}$ drops from $I(\lambda, \mathbf{x})$ giving the following observed light source color:

$$\mathbf{e} = \begin{pmatrix} e_R \\ e_G \\ e_B \end{pmatrix} = \int_{\omega} I(\lambda) \boldsymbol{\rho}(\lambda) d\lambda. \tag{2}$$

Knowing the direction of $\mathbf{e}$ is enough to perform a successful chromatic adaptation [2]. If only image pixel values

(a) (b)

Fig. 1: Same scene under different illuminations.

f are given without any other information, then calculating $\mathbf{e}$ is an ill-posed problems since both $I(\lambda)$ and $\rho_c(\lambda)$ are unknown. Therefore $\mathbf{e}$ is estimated by introducing additional assumptions, which has led to many illumination estimation methods that are generally divided in two groups. In the first group are low-level statistics-based methods like White-patch (WP) [3] [4], improved White-patch (IWP) [5], Gray-world (GW) [6], Shades-of-Gray (SoG) [7], Grey-Edge (1st and 2nd order (GE1 and GE2)) [8], Weighted Gray-Edge [9], using bright pixels (BP) [10], Color Sparrow (CS) [11], Color Rabbit (CR) [12], using color distribution (CD) [13], while the second group consists of methods that perform learning like gamut mapping (pixel, edge, and intersection based - PG, EG, and IG) [14], using neural networks [15], using high-level visual information (HLVI) [16], natural image statistics (NIS) [17], Bayesian learning (BL) [18], spatio-spectral learning (maximum likelihood estimate (SL), and with gen. prior (GP)) [19], Color Cat [20], Color Dog [21].

The statistics-based methods are faster, but the learning-based methods are more accurate. Models of many of learning-based methods rely on image features that are sometimes quite demanding to extract and process, which further results in impracticality for hardware implementation [22] despite a relatively high accuracy. In this paper new easily calculated features based on the histograms of the red chromaticity in the image are proposed. A new method based on these features is proposed and the accuracy of an existing one is improved to obtain state-of-the-art accuracy in both cases.

The paper is structured as follows: In Section II a brief description of some properties of the red chromaticity is given, in Section III it is used to propose both a new illumination estimation method and an improvement to an existing one,

in Section IV the experimental results are presented and discussed, and Section V concludes the paper.

II. Red chromaticity

Color is in the center of illumination estimation methods. Even though some methods rely on spatial distribution of pixels of a certain color, e.g. the Gray-edge method [8], it has been shown that mere color distribution is enough for a successful illumination estimation [13]. In most illumination estimation methods color is described by using its coordinates i.e. components in the RGB colorspace: red, green, and blue. If the amplitude of the color vector is not important, as is the case with $\mathbf{e}$, chormaticity can be used instead of the full color description. The components of the often used L^1-norm chromaticity for the RGB color are given as

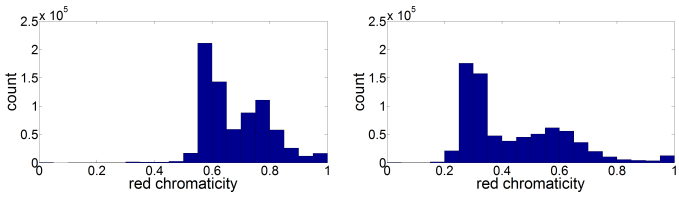
$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}. \quad (3)$$

Only two components are enough since the third one can be recovered by using the equation $r + g + b = 1$.

Under a reddish illumination the image pixels also tend to be more reddish, which is reflected on red chromaticity values of their pixels as shown in Fig .2 and Fig. 3.

(a) (b)

Fig. 2: Example of using the Gray-world method for illumination estimation: (a) the original image and (b) the chromatically adapted image.



(a) (b)

Fig. 3: Histograms of red chromaticity pixel values for images shown in Fig. 2.

One of the ways to investigate whether there is a useful connection between red chromaticity found in the image and the color of image scene illumination is to calculate their correlation. To do that, the images from the GreyBall dataset were used together with the ground-truth illumination that comes with them. First, for each image the chromaticity components of the ground-truth illumination have been calculated. Then for each image and for every value of $n \in \{2, ..., 20\}$ a histogram of the red chromaticity with n bins has been calculated. At the end, for each of the 209 resulting bins the correlation between its value across the images and the red chromaticity components of the ground-truth illumination has been calculated. The sorted absolute values of correlations of all these bins are shown in Fig. 4. Similar and even better results can be obtained for other datasets. It can be seen that for certain bins the correlation is relatively high, which leads to the idea that this high correlation could somehow be used in estimating the red chromaticity component of the illumination. From this estimation, the blue chromaticity component can also be accurately estimated as shown in [20] and from these two, the green one as well. The question that remains is how to perform this regression by using red chromaticity histograms?

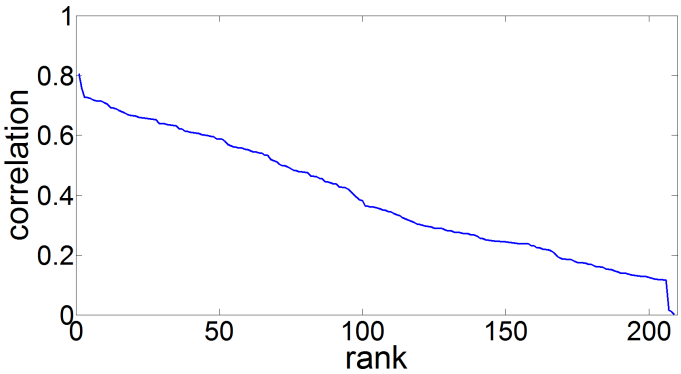


Fig. 4: Sorted absolute values of correlations between individual histogram bins of image red chromaticity values and the red chromaticity component of the ground-truth illumination for the GreyBall dataset [23].

III. Proposed application

A. Color Ant

One of the simplest algorithms that can be applied for the regression task is the k-nearest neighbor (k-NN) algorithm [24]. Using k-NN has already been proposed for illumination estimation in [25] where features are the results of other methods and the distance function is defined as the sum of angles between different parts of the feature vectors. Nevertheless, the accuracy results were not especially good.

Here we simply propose to use k-NN with red chromaticity histograms as features. To keep it simple, for the distance measure we propose to use the Euclidean distance between the feature vectors. The feature space is filled with feature vectors of images with previously known ground-truth illumination vectors $\mathbf{g} = (g_R, g_G, g_B)^T$. When a feature vector for a new given image is calculated, first k of its nearest neighbors i.e. images are found in the feature space and for each of them their ground-truth illumination is taken for a total of k illumination vectors. For red, green, and blue components of these k vectors the median value is calculated and the resulting three median values are combined into a single vector, which represents the illumination estimation of the given image. The median is used instead of mean or some other statistics in order to reduce the sensitivity to outliers. This whole approach has two hyperparameters: the resolution of the red chromaticity histogram n and the number of nearest neighbors k. The values

TABLE I: Angular error of selected low-level statistics-based methods, the proposed method, and selected learning-based methods on nine NUS benchmark image databases (lower is better).

	Low-level statistics-based methods							**Learning-based methods**									
Method	CR	CD	GW	WP	GGW	GE1	GE2	CA	SCC	CD_{SCC}	$\text{CD}_{GW,WP}$	PG	EG	IG	ML	GP	NIS
Dataset	**Mean angular error (°)**																
Canon1	3.09	**2.93**	5.16	7.99	3.16	3.45	3.47	3.10	3.34	3.16	3.13	6.13	6.07	6.37	3.58	3.21	4.18
Canon2	2.81	2.81	3.89	10.96	3.24	3.22	3.21	3.31	3.18	3.21	2.83	14.51	15.36	14.46	2.80	**2.67**	3.43
Fuji	**2.94**	3.15	4.16	10.20	3.42	3.13	3.12	3.60	5.95	5.36	3.36	8.59	7.76	6.80	3.12	2.99	4.05
Nikon1	3.06	**2.90**	4.38	11.64	3.26	3.37	3.47	2.93	3.65	3.27	3.19	10.14	13.00	9.67	3.22	3.15	4.10
Oly	2.65	2.76	3.44	9.78	3.08	3.02	2.84	2.59	3.49	3.59	**2.57**	6.52	13.20	6.21	2.92	2.86	3.22
Pan	2.89	2.96	3.82	13.41	3.12	2.99	2.99	2.90	3.23	3.26	**2.84**	6.00	5.78	5.28	2.93	2.85	3.70
Sam	2.94	**2.91**	3.90	11.97	3.22	3.09	3.18	3.01	2.92	2.98	2.92	7.74	8.06	6.80	3.11	2.94	3.66
Sony	2.88	2.93	4.59	9.91	3.20	3.35	3.36	**2.61**	3.07	3.04	2.83	5.27	4.40	5.32	3.24	3.06	3.45
Nikon2	3.57	3.81	4.60	12.75	4.04	3.94	3.95	3.70	4.08	4.11	**3.37**	11.27	12.17	11.27	3.80	3.59	4.36
Dataset	**Median angular error (°)**																
Canon1	2.08	2.01	4.15	6.19	2.35	2.48	2.44	1.94	2.55	**1.67**	1.72	4.30	4.68	4.72	2.80	2.67	3.04
Canon2	1.86	1.89	2.88	12.44	2.28	2.07	2.29	1.93	2.34	1.95	**1.85**	14.83	15.92	14.72	2.32	2.03	2.46
Fuji	1.84	2.15	3.30	10.59	2.60	1.99	2.00	1.98	2.66	2.60	**1.81**	8.87	8.02	5.90	2.70	2.45	2.95
Nikon1	**1.91**	2.08	3.39	11.67	2.31	2.22	2.19	2.10	2.50	2.00	1.94	10.32	12.24	9.24	2.43	2.26	2.40
Oly	1.79	1.87	2.58	9.50	2.15	2.11	2.18	1.71	2.24	1.74	**1.46**	4.39	8.55	4.11	2.24	2.21	2.17
Pan	1.70	2.02	3.06	18.00	2.23	2.16	2.04	1.73	2.15	**1.62**	1.69	4.74	4.85	4.23	2.28	2.22	2.28
Sam	1.88	2.03	3.00	12.99	2.57	2.23	2.32	1.97	2.00	**1.76**	1.89	7.91	6.12	6.37	2.51	2.29	2.77
Sony	2.10	2.33	3.46	7.44	2.56	2.58	2.70	**1.73**	2.17	1.90	1.77	4.26	3.30	3.81	2.70	2.58	2.88
Nikon2	2.42	2.72	3.44	15.32	2.92	2.99	2.95	2.45	2.58	2.23	**2.12**	10.99	11.64	11.32	2.99	2.89	3.51
Dataset	**Trimean angular error (°)**																
Canon1	2.56	2.22	4.46	6.98	2.50	2.74	2.70	2.14	2.66	2.05	**2.08**	4.81	4.87	5.13	2.97	2.79	3.30
Canon2	2.17	2.12	3.07	11.40	2.41	2.36	2.37	2.39	2.53	2.26	**2.07**	14.78	15.73	14.80	2.37	2.18	2.72
Fuji	**2.13**	2.41	3.40	10.25	2.72	2.26	2.27	2.35	3.01	3.00	2.20	8.64	7.70	6.19	2.69	2.55	3.06
Nikon1	2.23	2.19	3.59	11.53	2.49	2.52	2.58	2.22	2.76	2.26	**2.14**	10.25	11.75	9.35	2.59	2.49	2.77
Oly	2.01	2.05	2.73	9.54	2.35	2.26	2.20	1.86	2.49	2.15	**1.72**	4.79	10.88	4.63	2.34	2.28	2.42
Pan	2.12	2.31	3.15	14.98	2.45	2.25	2.26	1.98	2.31	1.99	**1.87**	4.98	5.09	4.49	2.44	2.37	2.67
Sam	2.18	2.22	3.15	12.45	2.66	2.32	2.41	2.16	2.27	**2.03**	2.05	7.70	6.56	6.40	2.63	2.44	2.94
Sony	2.26	2.42	3.81	8.78	2.68	2.76	2.80	**1.88**	2.35	2.21	2.03	4.45	3.45	4.13	2.82	2.74	2.95
Nikon2	2.67	3.10	3.69	13.80	3.22	3.21	3.38	2.60	2.85	2.59	**2.38**	11.11	12.01	11.30	3.11	2.96	3.84

of these parameters are obtained in the model selection [26] by means of cross-validation during the training phase.

Since the proposed method relies on a kind of swarm, and so do ants, it was named Color Ant (CA). The pseudocode for applying Color Ant is given in Algorithm 1. For good performance the P points representing images with known illuminations can be organized using a k-d tree [27], which leads to a $O(k \log P)$ complexity for finding the nearest neighbors. Since the median that is further needed can be found in $O(k)$ complexity [28], the overall complexity is $O(k(1 + \log P) + N)$ where N is the number of pixels.

Algorithm 1 Color Ant

1: $I = GetImage()$
2: $\mathbf{h} = I.CalculateRedChromaticityHistogram(n)$
3: **for** $i = 1$ **to** k **do**
4: $\quad \mathbf{g}^{(i)} = GetNeighborIllumination(\mathbf{h}, i)$
5: **end for**
6: $\mathbf{e} = \left(\underset{i \in \{1,\ldots,k\}}{\text{median}} \{g_R^{(i)}\}, \underset{i \in \{1,\ldots,k\}}{\text{median}} \{g_G^{(i)}\}, \underset{i \in \{1,\ldots,k\}}{\text{median}} \{g_B^{(i)}\} \right)^T$

B. Smart Color Cat

Color Cat is a recently proposed global learning-based illumination estimation method [20]. It relies on normalized image color histograms with n^3 bins of equal size where n is the resolution across a single axis in the 3-D RGB colorspace. The histogram bins are first transformed by multiplying them with a matrix $\mathbf{M}$ to extract the principal components. These components are then used as features for linear regression with coefficients $\mathbf{c}$ to get $x \in [0, 1]$. This is the position of the red chromaticity of illumination between values r_0 and r_1 that are previously learned as likely extreme values. By exploiting the fact that real world illumination values that can be modelled with a line, from the red chromaticity illumination component the blue one is calculated as $b = a_1 r + a_0$, where a_1 and a_0 are line parameters. From these two the green component is calculated by utilizing the fact that the sum of all three chromaticity components is 1. For more detail see [20].

Even though Color Cat was shown to give high accuracy results on large datasets, the training process is relatively long due to the large number of bins that are processed during the model selection for obtaining the optimal value of n and the optimal number of principal components k that are used in linear regression. Since n^3 grows fast with n, another problem is that training sets with a smaller number of images offer a relatively low upper bound for n that may be used because for linear regression the number of images should be greater than the number of bins. This constraint prevents learning from being well performed if the number of images in the training set is not big and this consequently leads to lower accuracy.

As it was explained, the main source of the problems is the fast growth of the number of color histogram bins n^3. In the light of the previously discussed red chromaticity histograms, a solution to this problem might be to simply use them as features instead. With them the number of bins is n, which is significantly lower then with color histograms. Since with this modification the histograms are more practical

and enable a faster training and application phase, the new method obtained by slightly changing the original Color Cat in a smart way was therefore named Smart Color Cat (SCC). The pseudocode for applying Smart Color Cat is given in Algorithm 2. In comparison to Color Cat complexity, which is $O\left(k(n^3+1)+N\right)$ where N is the number of image pixels, the complexity of Smart Color Cat is $O\left(k(n+1)+N\right)$.

Algorithm 2 Smart Color Cat

1: $I = GetImage()$
2: $\mathbf{h} = I.CalculateRedChromaticityHistogram(n)$
3: $\mathbf{h}' = \mathbf{Mh}$
4: $x = \mathbf{c}^T\mathbf{h}'$
5: $r = x(r_1 - r_0) + r_0$
6: $b = a_1 r + a_0$
7: $g = 1 - r - b$
8: $\mathbf{e}=(r, g, b)^T$

TABLE II: Performance of different color constancy methods on the original GreyBall dataset (lower is better).

method	**mean** (°)	**median** (°)	**trimean** (°)
do nothing	8.28	6.70	7.25
Low-level statistics-based methods			
GW	7.87	6.97	7.14
WP	6.80	5.30	5.77
SoG	6.14	5.33	5.51
general GW	6.14	5.33	5.51
GE1	5.88	4.65	5.11
GE2	6.10	4.85	5.28
Learning-based methods			
PG	7.07	5.81	6.12
EG	6.81	5.81	6.03
IG	6.93	5.80	6.05
NIS	5.19	3.93	4.31
EB	4.38	3.43	3.67
CC	**4.22**	3.17	**3.46**
SCC	4.62	3.52	3.80
CA	5.04	3.73	4.10
$\text{CD}_{WP,GW}$	5.27	3.71	4.16
CD_{CC}	4.50	**2.86**	3.50
CD_{CA}	5.47	4.17	4.52
CD_{SCC}	4.80	3.08	3.71

IV. Experimental results

A. Benchmark datasets

For the purpose of testing the accuracy of the proposed method and the proposed improvement, several benchmark datasets have been used: the GreyBall dataset [23], its approximated linear version, and the nine linear NUS dataset [13]. The well-known linear version of the ColorChecker dataset [18] [29] was not used because in many publications it has been used in a wrong way by not subtracting the dark level [30], which leads to wrong estimations and it complicates the accuracy comparison. Since the model used in Eq. (1) is linear, linear datasets are a better choice than non-linear ones. Additionally, the images on which illumination estimation is performed in digital devices are also linear [31].

TABLE III: Performance of different color constancy methods on the linear GreyBall dataset (lower is better).

method	**mean** (°)	**median** (°)	**trimean** (°)
do nothing	15.62	14.00	14.56
Low-level statistics-based methods			
GW	13.01	10.96	11.53
WP	12.68	10.50	11.25
SoG	11.55	9.70	10.23
general GW	11.55	9.70	10.23
GE1	10.58	8.84	9.18
GE2	10.68	9.02	9.40
Learning-based methods			
PG	11.79	8.88	9.97
EG	12.78	10.88	11.38
IG	11.81	8.93	10.00
HVLI	9.73	7.71	8.17
NIS	9.87	7.65	8.29
EB	**7.97**	6.46	6.77
CC	8.73	7.07	7.43
SCC	8.18	6.28	6.73
CA	8.93	6.51	7.07
$\text{CD}_{WP,GW}$	10.27	7.33	8.20
CD_{CC}	8.81	5.98	6.97
CD_{CA}	9.21	6.15	7.20
CD_{SCC}	8.51	**5.55**	**6.56**

In the datasets each of the images is accompanied by its ground-truth illumination, which has been measured by placing a calibration object in the image scene, e.g. a grey ball or a color checker. Before the image is used to test an illumination estimation method, the calibration object has to be masked out to prevent any possible bias on the tested method.

B. Training and testing

Even though there are several accuracy measures for illumination estimation for a single image [32] [33] [34], the most widely used one is the angle between the ground-truth illumination vector and the illumination estimation vector. When these angles i.e. angular errors are calculated for all images in a given dataset, they are usually described by using various statistics. The most important of these statistics is the median angular error [35]. Median is preferred to mean because of the non-symmetry of the angular error distribution.

For both Color Ant and Smart Color Cat the number of histogram bins n is a hyperparameter whose optimal value is determined during the model selection. In the performed tests the value of n was chosen from the set $\{2, ..., 20\}$. Since model selection contains cross-validation, the whole testing process was performed by means of nested cross-validation [26]. For the GreyBall dataset the outer loop performed a 15-fold cross-validation with the folds provided by the dataset authors, while for the NUS datasets the outer loop performed a 3-fold cross-validation. As for the inner loop, for both the GreyBall dataset and the NUS datasets it performed a 5-fold cross-validation. The source code used for training and using both the Color Ant and the Smart Color Cat method is publicly available at http://www.fer.unizg.hr/ipg/resources/color_constancy/.

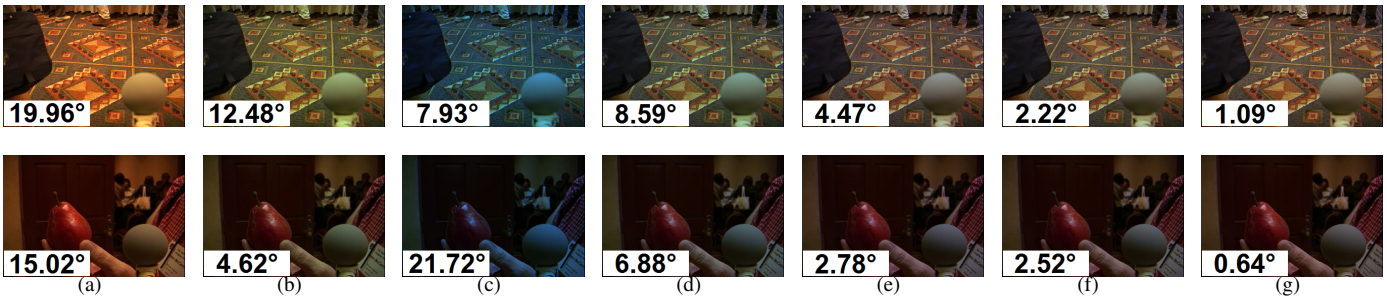


Fig. 5: Examples of chromatic adaptation based on the methods' illumination estimation and respective illumination estimation errors: (a) do nothing, (b) White-patch, (c) Gray-world, (d) Exemplar-based, (e) Color Cat), (f) Color Ant, and (g) Smart Color Cat.

C. Accuracy

Table I, Table II, and Table III show some of the angular error statistics obtained by various illumination estimation methods for the NUS datasets, GreyBall dataset, and its linear version, respectively. The results were taken from several sources [13] [31] [36]. For the linear NUS datasets Color Ant and Smart Color Cat perform close to the state-of-the-art results. The Smart Color Cat performs even better when used as a voter for the Color Dog method (CD_{SCC}) so that in some cases it outperforms the Color Dog method with White-patch and Gray-world methods as voters. Since the accuracy of Color Ant did not change significantly when it was used as a Color Dog voter, the results of this setup were not shown.

As for the GreyBall dataset and its linear version, Color Ant was shown to be highly accurate given its relatively simple model. This can be attributed to the non-linear nature of the k-NN algorithm, which is in the core of Color Ant. Smart Color Cat used as a Color Dog voter achieves very accurate results when compared to other methods. For the GreyBall dataset it is outperformed only by the regular Color Cat, but for the linear GreyBall dataset, it is significantly better then all other methods. This shows how well suited the Smart Color Cat model is for non-linear images or approximated linear images like the ones in the GreyBall dataset. Fig. 5 shows examples of chromatic adaptation based on different methods' illumination estimation and respective illumination estimation errors.

An additional question that arises from the given results is whether Smart Color Cat is better than Color Cat or is it vice-versa? When looking at the results obtained on the GreyBall dataset and its linear version, both methods have a similar performance. However, Smart Color Cat's smaller histograms enable a significantly better learning process on smaller datasets, e.g. the NUS datasets, because the histogram resolution n can take higher values. This leads to Smart Color Cat obtaining higher accuracy then Color Cat on such datasets and to a much lower learning time. Out of these reasons in the general case Smart Color Cat is better than Color Cat.

D. Other chromaticities

Beside red chromaticity histograms, another possibility might have been green or blue chromaticity histograms. However, experiments with these histograms gave a significantly lower accuracy and therefore these results were not shown.

V. Conclusion

New features for global illumination estimation have been proposed. Beside being easy to calculate, these features enable a new accurate illumination estimation method and they were also shown to successfully improve the accuracy of the Color Cat method and simplify its training and application phase. For some of the used datasets the accuracy of the improved Color Cat method was therefore higher than the accuracy of any other method. Surprisingly, although the other newly proposed method is relatively simple, it was shown to compare very well with the state-of-the-arts methods thus further demonstrating the power of the new features. In future other possible applications of the proposed features should be examined.

References

[1] M. Ebner, *Color Constancy*, ser. The Wiley-IS&T Series in Imaging Science and Technology. Wiley, 2007.

[2] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data," *Image Processing, IEEE Transactions on*, vol. 11, no. 9, pp. 972–984, 2002.

[3] E. H. Land, *The retinex theory of color vision*. Scientific America., 1977.

[4] B. Funt and L. Shi, "The rehabilitation of MaxRGB," in *Color and Imaging Conference*, vol. 2010, no. 1. Society for Imaging Science and Technology, 2010, pp. 256–259.

[5] N. Banić and S. Lončarić, "Improving the White patch method by subsampling," in *Image Processing (ICIP), 2014 21st IEEE International Conference on*. IEEE, 2014, pp. 605–609.

[6] G. Buchsbaum, "A spatial processor model for object colour perception," *Journal of The Franklin Institute*, vol. 310, no. 1, pp. 1–26, 1980.

[7] G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Color and Imaging Conference*, vol. 2004, no. 1. Society for Imaging Science and Technology, 2004, pp. 37–41.

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

[9] A. Gijsenij, T. Gevers, and J. Van De Weijer, "Improving color constancy by photometric edge weighting," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 918–929, 2012.

[10] H. R. V. Joze, M. S. Drew, G. D. Finlayson, and P. A. T. Rey, "The Role of Bright Pixels in Illumination Estimation," in *Color and Imaging Conference*, vol. 2012, no. 1. Society for Imaging Science and Technology, 2012, pp. 41–46.

[11] N. Banić and S. Lončarić, "Using the Random Sprays Retinex Algorithm for Global Illumination Estimation," in *Proceedings of The Second Croatian Computer Vision Workshopn (CCVW 2013)*, S. Lončarić and

S. Šegvić, Eds., no. 1. University of Zagreb Faculty of Electrical Engineering and Computing, Sep. 2013, pp. 3–7.

[12] ——, "Color Rabbit: Guiding the Distance of Local Maximums in Illumination Estimation," in *Digital Signal Processing (DSP), 2014 19th International Conference on*. IEEE, 2014, pp. 345–350.

[13] 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.

[14] G. D. Finlayson, S. D. Hordley, and I. Tastl, "Gamut constrained illuminant estimation," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 93–109, 2006.

[15] V. C. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity by using a neural network," *JOSA A*, vol. 19, no. 12, pp. 2374–2386, 2002.

[16] J. Van De Weijer, C. Schmid, and J. Verbeek, "Using high-level visual information for color constancy," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[17] A. Gijsenij and T. Gevers, "Color Constancy using Natural Image Statistics." in *CVPR*, 2007, pp. 1–8.

[18] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[19] A. Chakrabarti, K. Hirakawa, and T. Zickler, "Color constancy with spatio-spectral statistics," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 8, pp. 1509–1519, 2012.

[20] N. Banić and S. Lončarić, "Color Cat: Remembering Colors for Illumination Estimation," *Signal Processing Letters, IEEE*, vol. 22, no. 6, pp. 651–655, 2015.

[21] ——, "Color Dog: Guiding the Global Illumination Estimation to Better Accuracy," in *VISAPP*, 2015, pp. 129–135.

[22] I. Harbaš, N. Banić, D. Jurić, S. Lončarić, and M. Subašić, "Computer vision-based advanced driver assistance systems," in *34th Conference on Transportation Systems with International Participation AUTOMATION IN TRANSPORTATION 2014*, 2014.

[23] F. Ciurea and B. Funt, "A large image database for color constancy research," in *Color and Imaging Conference*, vol. 2003, no. 1. Society for Imaging Science and Technology, 2003, pp. 160–164.

[24] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[25] Y. Jia-zheng, T. Li-yan, B. Hong, H. Jing-hua, and Z. Rui-zhe, "Llumination estimation combining physical and statistical approaches," in *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, vol. 1. IEEE, 2009, pp. 365–368.

[26] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

[27] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[28] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan, "Linear time bounds for median computations," in *Proceedings of the fourth annual ACM symposium on Theory of computing*. ACM, 1972, pp. 119–124.

[29] B. F. L. Shi. (2015, May) Re-processed Version of the Gehler Color Constancy Dataset of 568 Images. [Online]. Available: http://www.cs.sfu.ca/colour/data/

[30] S. E. Lynch, M. S. Drew, and k. G. D. Finlayson, "Colour Constancy from Both Sides of the Shadow Edge," in *Color and Photometry in Computer Vision Workshop at the International Conference on Computer Vision*. IEEE, 2013.

[31] A. Gijsenij, T. Gevers, and J. Van De Weijer, "Computational color constancy: Survey and experiments," *Image Processing, IEEE Transactions on*, vol. 20, no. 9, pp. 2475–2489, 2011.

[32] A. Gijsenij, T. Gevers, and M. P. Lucassen, "Perceptual analysis of distance measures for color constancy algorithms," *JOSA A*, vol. 26, no. 10, pp. 2243–2256, 2009.

[33] G. D. Finlayson and R. Zakizadeh, "Reproduction angular error: An improved performance metric for illuminant estimation," *perception*, vol. 310, no. 1, pp. 1–26, 2014.

[34] N. Banić and S. Lončarić, "A Perceptual Measure of Illumination Estimation Error," in *VISAPP*, 2015, pp. 136–143.

[35] S. D. Hordley and G. D. Finlayson, "Re-evaluating colour constancy algorithms," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1. IEEE, 2004, pp. 76–79.

[36] T. G. A. Gijsenij and J. van de Weijer. (2015, May) Color Constancy — Research Website on Illuminant Estimation. [Online]. Available: http://colorconstancy.com/