Color Constancy Using Standard Deviation of Color Channels

Anustup Choudhury and Gérard Medioni
Department of Computer Science
University of Southern California
Los Angeles, California 90089. USA
{achoudhu,medioni}@usc.edu

***Abstract*—We address here the problem of color constancy and propose a new method to achieve color constancy based on the statistics of images with color cast. Images with color cast have standard deviation of one color channel significantly different from that of other color channels. This observation is also applicable to local patches of images and ratio of the maximum and minimum standard deviation of color channels of local patches is used as a prior to select a pixel color as illumination color. We provide extensive validation of our method on commonly used datasets having images under varying illumination conditions and show our method to be robust to choice of dataset and at least as good as current state-of-the-art color constancy approaches.**

***Keywords*-color; color constancy; illumination;**

I. Introduction

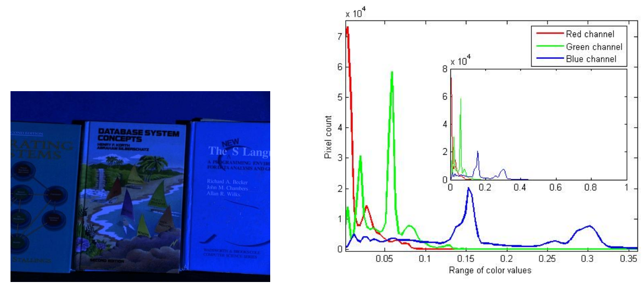
Color Constancy is a phenomenon that describes the human ability to estimate the actual color of a scene irrespective of the color of illumination of that scene. Since an image is a product of the illumination that falls on the scene and the reflectance properties of the scene, achieving color constancy is an ill-posed problem and various techniques have been proposed to address it.

Our method is based on the observation that an image of a scene, taken under colored illumination, has one color channel that has significantly different standard deviation from at least one other color channel. Figure 1(a) has a strong blue color cast and the standard deviations of the RGB color channels are $\sigma_R = 0.0184$, $\sigma_G = 0.0267$ and $\sigma_B = 0.0941$. We can see that the value of σ_B is 5 times more than σ_R. If we remove the color cast from Figure 1(a) as shown in Figure 2(a), the standard deviations of the RGB color channels are $\sigma_R = 0.0591$, $\sigma_G = 0.0500$ and $\sigma_B = 0.0582$. We observe that the standard deviations of the color channels of an image with no color cast are very similar to each other. We find the ratio of the maximum and minimum standard deviation of color channels of local patches of an image and use that as a prior to estimate the color of illumination and achieve color constancy.

II. Previous Work

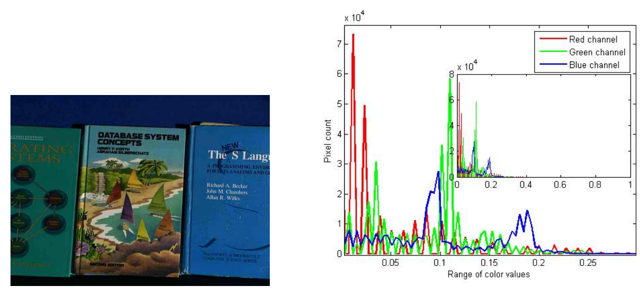
Any acquired image, I can be represented as:

$$I = \int_{\omega} l(\lambda)s(\lambda)c(\lambda)d\lambda, \tag{1}$$



(a) (b)

Figure 1. (a) Image from [1] with blue color cast. The intensity is tripled for better clarity. (b) Zoomed-in RGB histogram of Figure 1(a) with original intensity. The complete histogram is shown in the inset of Figure 1(b)



(a) (b)

Figure 2. (a) Image from original Figure 1(a) without color cast. The intensity is tripled for better clarity. (b) Zoomed-in RGB histogram of Figure 2(a) with original intensity. The complete histogram is shown in the inset of Figure 2(b)

where, ω is the visible spectrum, $l(\lambda)$ is the spectral distribution of illuminance, $s(\lambda)$ is the spectral reflectance and $c(\lambda)$ is the camera sensitivity to wavelength λ. Color constancy algorithms make an assumption that only one light source is used to illuminate the scene. As the observed illumination color depends on the actual illumination color and the camera property, achieving color constancy is equivalent to

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.426

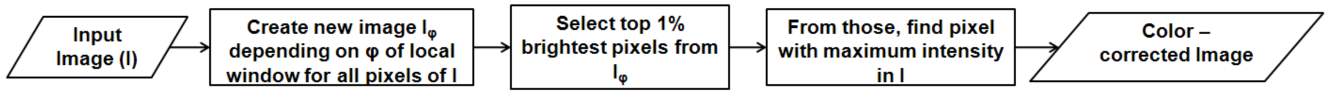


Figure 3. Flowchart of our method

estimating l:

$$l = \int_\omega l(\lambda)c(\lambda)d\lambda, \tag{2}$$

given the color values of $I(x, y)$ where (x, y) are the pixel co-ordinates of I.

Many algorithms have been proposed to achieve color constancy using low-level features such as, White-Patch Assumption [2] where the maximum pixel value is assumed to be white, the Grey-World algorithm [3] where the average pixel value is grey and the Grey-Edge algorithm [4] where higher order derivative of the image is used. As shown in [4], all the above techniques can be expressed as:

$$\left(\int \left| \frac{\partial^n i^\sigma(x)}{\partial x^n} \right|^p dx \right)^{\frac{1}{p}} = kl^{n,p,\sigma}, \tag{3}$$

where, n is the order of derivative, p is the Minkowski norm and σ is the parameter for smoothing the image i with a Gaussian filter.

The most recent techniques combine [2], [3] and [4] depending on different criteria. Gijsenij and Gevers [5] use Weibull parameterization to get the characteristics of the image and, depending on those values, divide the image space into clusters using k-means algorithm and then use the best color constancy algorithm corresponding to that cluster. The best algorithm for a cluster is learnt from the training dataset. 3D scene geometry is used to classify images and a color constancy algorithm is chosen according to the classification results to estimate the illuminant color [6]. Other more complex algorithms include "Beyond Bags of Pixels approach" [7] where spatial dependencies between the pixels of the image are considered. Cardei *et al.* [8] use a neural network to learn the illumination of a scene from a large number of training data. A nonparametric linear regression tool called kernel regression has also been used to estimate the illuminant chromaticity [9]. Finlayson *et al.* [10] use the knowledge about appearance of colors under a certain illumination as a prior to estimate the probability of an illuminant from a set of illuminations. The disadvantage of this method is that the estimation of the illuminant depends on a good model of the lights and surfaces, which is not easily available. GCIE (Gamut-constraied illumination estimation) method tries to estimate the illuminant color by finding an appropriate mapping from an image gamut to a *canonical* gamut and constrains the transformations so that the illuminant estimation corresponds to a pre-defined set of illuminants [11].

III. Our Approach

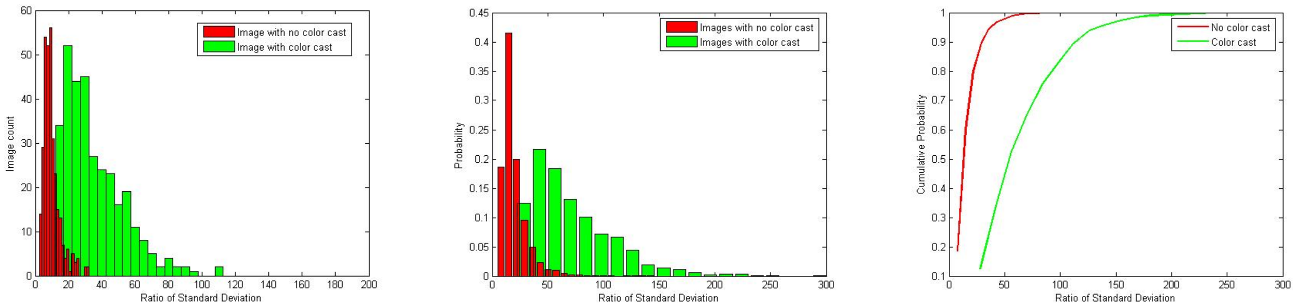
As described in Section I, for images with color cast, the standard deviation of one color channel is significantly different from that of other color channels. This can be characterized by the ratio between $\sigma_{\max} = \max\{\sigma_i, i \in R, G, B\}$ and $\sigma_{\min} = min\{\sigma_i, i \in R, G, B\}$, where σ_i is the standard deviation of the color channel i. The value of this ratio $\phi = \sigma_{\max}/\sigma_{\min}$ will be very high for images with color cast and low for images without color cast. We find that, in most images under white illumination (without color cast), local patches of an image have similar standard deviations in all 3 color channels, and it is not the case for images with color cast. This leads us to believe that the change in standard deviation for those local patches is mainly contributed by the colored illumination. Therefore, we use information from these patches to select pixels to estimate the color of illumination.

Our method illustrated in Figure 3, consists of 2 key steps: 1) Create a new image I_ϕ, which is the ϕ value of local window of original image 2) Use the brightest pixels from I_ϕ as prior to select pixel from original image as illumination color. We create a new image with same resolution as the original image where every pixel of the new image is the ϕ value of a local window around the corresponding pixel in the original image. This can be formulated as follows:

$$I_\phi(x) = \forall_{x \in I} \frac{\max_{i \in \{R,G,B\}, \hat{y} \in W(x)}(\sigma_i(\hat{y}))}{\min_{i \in \{R,G,B\}, \hat{y} \in W(x)}(\sigma_i(\hat{y}))}, \tag{4}$$

where, i is the color channel R, G, B, $\hat{y}$ is the set of pixels in a window W centered at pixel x in the original image I. A lucid explanation of Equation 4 is that for every pixel x in image I, a window W ($\frac{W+1}{2} - 1$ pixels on either side of the current pixel) is considered around that pixel. For all pixels inside W, represented by $\hat{y}$, the standard deviation of all 3 color channels is calculated and the ratio of the maximum to the minimum standard deviation is used to create the image I_ϕ.

We use the controlled indoor environment [1] to verify how good I_ϕ is. This dataset has 30 different scenes under 11 different illumination conditions. Several images from this dataset were found unusable by the original authors [1] resulting in a dataset of 321 images. All images have the same resolution - 637 X 468. All images have been illuminated by just one source. The ground truth values of the illumination are provided. Figure 1(a) is an example from this dataset. The ground truth values of the illumination are already provided. These values are normalized by the



(a) (b) (c)

Figure 4. Statistics for images with and without color cast. Image (a) is the histogram of average ϕ. Image (b) is the histogram of ϕ over all images. Image (c) is the corresponding cumulative distribution

Euclidean norm of the illumination color vector. We use the normalized values to remove color cast from every image to create a new dataset as follows:

$$\text{Color Corrected Image} = \frac{1}{\sqrt{3}} \frac{\text{Original Image}}{C}, \quad (5)$$

where, C is the normalized color of illumination and $\sqrt{3}$ is a normalization constant based on the diagonal model to preserve the intensity of pixels.

We compute the average ϕ for every image and plot the corresponding histogram in Figure 4(a). From Figure 4(a) we can see that the average value of ϕ for images with color cast is much higher than the average value of ϕ for images without color cast. Figure 4(b) is the histogram of ϕ over all the images for both images with and without color cast and Figure 4(c) is the corresponding cumulative histogram. We can see that about 80% of the images without color cast have $\phi < 20$. On the other hand, for images with color cast, less than 12% of images have $\phi < 30$ and around 90% of images have $\phi < 125$. This gives us a very strong statistical support to use the value of ϕ to distinguish between images with and without color cast.

In case of White-Patch algorithm, the pixel with the highest intensity is assumed to be the color of illumination. But in images from the real world environment, due to noise or specular reflections, this assumption can be violated. In order to improve the estimation of the illumination color we use the reconstructed I_ϕ image from which we pick the top 1% brightest pixels. Among these pixels, the pixel with the highest intensity in the original input image I is selected as the illuminant color. It should be noted that this pixel need not be the brightest pixel in the image.

Once we have estimated the color of illumination, the color-corrected images as shown in Figure 5 can be obtained as shown in Equation 5. The images are corrected to how it would appear under a white illuminant.

There are certain limitations of our method. Similar to existing approaches, our proposed technique estimates only one illumination color and would give improper results for scenes illuminated with multiple light sources. In that case, the estimated color may be a combination of the different illumination colors. In the extreme scenario, if all the local patches of the image have the same value of ϕ, selection of the top 1% brightest pixels in I_ϕ image will be a problem. In that case, we can skip that step and directly use the White-Patch assumption on I. We believe that the performance of our method is bounded by the performance of the White-Patch assumption, as also can be seen later in Figure 6.

IV. Experiments and Results

In order to evaluate our color constancy algorithm, we conduct experiments on two widely used datasets. The first dataset [1] is described in Section III. The ground truth values of the illumination are known. The second dataset consists of 11000 images from 15 different scenes taken in a real world environment [12]. We randomly select 10 images from each scene. The ground truth illuminant value of the scene is computed from a grey ball that is present in the bottom right corner of every image as shown in Figure 5(a). The illumination color is available with the dataset and is used as ground truth in this experiment. While estimating the illumination color, that entire quadrant containing the grey ball is excluded as depicted by the white box in Figure 5(c) - Figure 5(f).

Angular error (in degrees) is used to measure the error between the estimated illumination color l_e and the ground truth illumination color l_{gt} and it can be computed as:

$$\text{angular error}, \epsilon = \cos^{-1}(\hat{l_e}.\hat{l_{gt}}), \quad (6)$$

where, $(\hat{.})$ stands for normalized values. The median of the angular error is then computed across the entire dataset [13].

The implementation of existing color constancy algorithms that uses low-level features were provided by the authors of [4] and the best parameters are chosen from [4]. The results from more complex color constancy algorithms such as color by correlation, gamut mapping and neural networks are reported in [14][8].

Some methods in the literature [9] have used the root mean square (RMS) error between the estimated illumination chromaticity, lc_e and the actual illumination chromaticity, lc_{gt} to evaluate their results although this is not the best metric [13]. Since we do not have access to the individual error values, we compare the results as is. The RMS_{rg} error can be calculated as:

$$RMS_{rg} = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \sum_{j=1}^{M} (lc_{e_i}^{j} - lc_{gt_i}^{j})^2 \right)^{1/2}, \quad (7)$$

where, N is the total number of images and M is the number of color channels (M = 2 for chromaticity space). We calculate error for the rg space. The chromaticity for r and g for the estimated illuminant can be computed as $lc_e^r = l_e^r/(l_e^r + l_e^g + l_e^b)$ and $lc_e^g = l_e^g/(l_e^r + l_e^g + l_e^b)$ where r, g and b are the color channels. Similarly, we can compute the chromaticity for the ground truth illuminant. The RMS_{rg} error for White-Patch, Neural Networks and Color by Correlation were presented in [14]. The performance of our method on the controlled indoor environment along with the best parameters is shown in Table I. W is the local window size for calculating I_ϕ.

Table I
ERROR FOR THE CONTROLLED INDOOR ENVIRONMENT.

Method	Parameters	Median ϵ (^{o})	RMS_{rg}
White-Patch	-	6.4	0.053
Grey World	-	6.9	-
1^{st} order Grey Edge	$p = 7$	3.2	-
2^{nd} order Grey Edge	$p = 7$	2.8	-
Color by Correlation	-	3.1	0.061
Gamut Mapping	-	2.9	-
Neural Networks	-	7.7	0.071
Kernel Regression	-	-	0.052
SVMs	-	-	0.066
Our Method	***W = 11 X 11***	***2.8***	***0.044***

All the results presented in the literature train and test on similar images. So, the parameters from [4] vary according to the dataset. In order to show robustness, we find the best parameters for the first dataset and use those parameters for the second dataset and still show performance gain. The performance of our method on the real world environment is shown in Table II along with the best parameters.

From Table II, we can see that our method has a 15.4% improvement over the Grey-edge algorithm. On comparing with a very recent technique - "Beyond Bag of pixels" approach [7], we find that our method gives us almost 18.6% improvement.

We implemented the algorithm in MATLAB in Windows XP environment on a PC with Xeon processor. For an image with size 637 X 468 pixels, it takes approximately 9 seconds.

Table II
MEDIAN ANGULAR ERROR (DEGREES) FOR THE REAL WORLD ENVIRONMENT.

Method	Parameters	Median
White-Patch	-	4.85
Grey World	-	7.36
1^{st} order Grey Edge	$p = 6$	4.41
Beyond Bags of Pixels	-	4.58
Our Method	***W = 11 X 11***	***3.73***

The effects of W on the median angular error for the controlled indoor environment dataset [1] can be seen in Figure 6 (Y-axis is inverted for better visualization). Small values of W have larger error because of insufficient information in the small local patches whereas as W increases, the median angular error will eventually converge to the error of White-Patch algorithm when W is the image size.

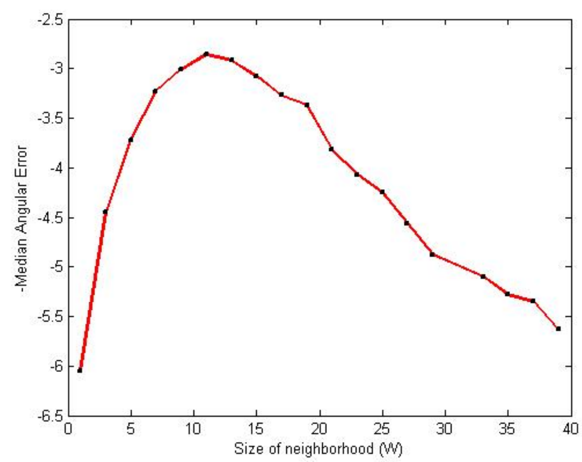
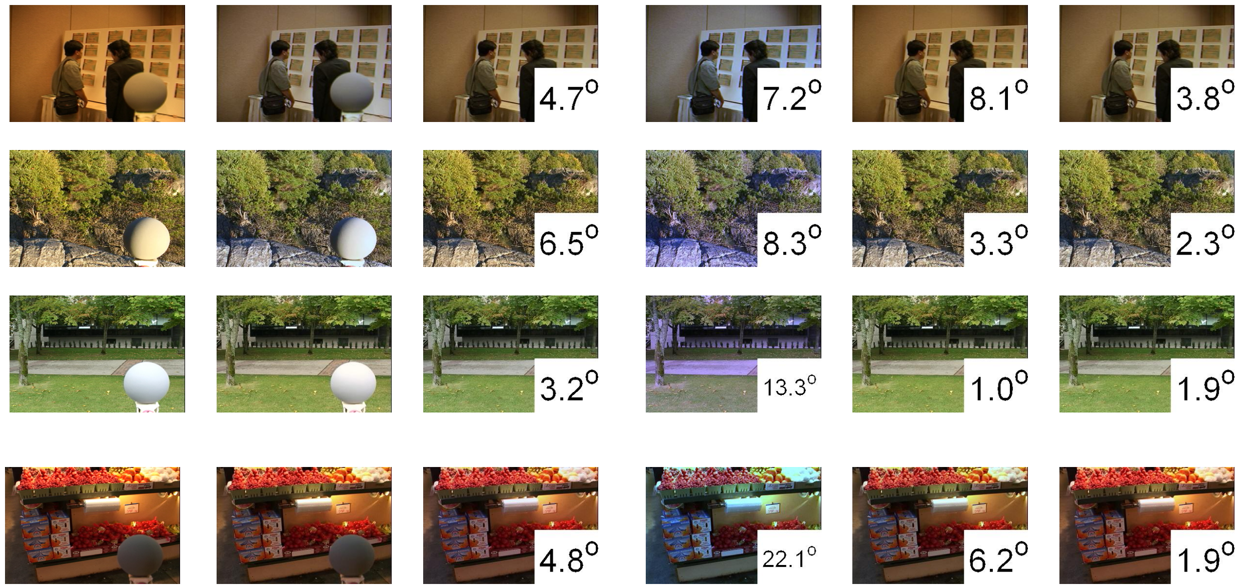


Figure 6. Effect of window size W on median angular error for the controlled indoor environment

V. CONCLUSION AND FUTURE WORK

We have proposed a new technique to achieve color constancy that is based on the statistics of images with color cast. The illumination estimation may not always be correct if noise is present as it may cause abnormal change in the ratio of standard deviations. Preprocessing with denoising algorithms will solve this problem. We conducted experiments on two widely used datasets and show that our method is robust to choice of dataset and gives results that are atleast as good as existing state-of-the-art color constancy methods.

Our observation of the intermediate I_ϕ image reveals that many pixels that were chosen were present along the edges in the image. As future work, it will be interesting to find out if our technique is, in any way, analogous to the Grey-Edge method. This is because the Grey-Edge method, while



(a) (b) (c) (d) (e) (f)

Figure 5. Example of images from real world environment and their angular errors. (a) are the original images and their corrections using (b) ground truth values of the illumination, (c) White-Patch assumption, (d) Grey-world algorithm, (e) Grey-edge algorithm and (f) Our method.

trying to estimate the illumination color, finds derivative of the image that can be considered equivalent to finding edges in an image.

ACKNOWLEDGMENT

This research was supported by the National Institutes of Health Grant EY016093.

REFERENCES

[1] K. Barnard, L. Martin, B. Funt, and A. Coath, "A data set for color research," *Color Research and Application*, vol. 27, no. 3, pp. 147–151, 2002.

[2] E. H. Land, "The retinex theory of color vision." *Scientific American*, vol. 237, no. 6, pp. 108–128, December 1977.

[3] G. Buchsbaum, "A spatial processor model for object dolor perception," *Franklin Inst.*, vol. 310, pp. 1–26, 1980.

[4] 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, 2007.

[5] A. Gijsenij and T. Gevers, "Color constancy using natural image statistics," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[6] R. Lu, A. Gijsenij, T. Gevers, D. Xu, V. Nedovic, and J. M. Geusebroek, "Color constancy using 3d stage geometry," in *IEEE International Conference on Computer Vision*, 2009.

[7] A. Chakrabarti, K. Hirakawa, and T. Zickler, "Color constancy beyond bags of pixels," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–6.

[8] V. C. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity by using a neural network," *Journal of the Optical Society of America*, vol. 19, no. 12, pp. 2374–2386, 2002.

[9] V. Agarwal, A. Gribok, A. Koschan, and M. Abidi, "Estimating illumination chromaticity via kernel regression," in *IEEE International Conference on Image Processing*, 2006, pp. 981–984.

[10] G. Finlayson, S. Hordley, and P. Hubel, "Color by correlation: A simple, unifying framework for color constancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1209–1221, November 2001.

[11] 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.

[12] F. Ciurea and B. V. Funt, "A large image database for color constancy research," in *Color Imaging Conference*, 2003, pp. 160–164.

[13] S. D. Hordley and G. D. Finlayson, "Re-evaluating colour constancy algorithms," in *International Conference on Pattern Recognition*, 2004, pp. 76–79.

[14] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms - part ii: Experiments with image data," *IEEE Transactions on Image Processing*, vol. 11, p. 2002, 2002.