

COLOR CONSTANCY USING DENOISING METHODS AND CEPSTRAL ANALYSIS

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 two new methods for achieving color constancy—the first method uses denoising techniques such as a Gaussian filter, Median filter, Bilateral filter and Non-local means filter to smooth the image for illuminant estimation, while the second method acts in the frequency domain by doing a cepstral analysis of the image.

We provide extensive validation tests for our illuminant estimation on commonly used datasets having images under different illumination conditions, and the results show that both new methods outperform current state-of-the-art color constancy approaches, at a very low computational cost.

Index Terms— Color constancy, denoising, cepstrum, illumination, retinex

1. INTRODUCTION

Any image can be expressed as a product of two components – the illumination image and the reflectance image. The Retinex Theory proposed by Land experimentally shows that the human visual system can identify the color of objects in a scene irrespective of the color of the illumination [16]. This phenomenon is called color constancy. Trying to recover the illumination is an under-constrained problem and various methods have been proposed to address it.

We introduce two new techniques to estimate the illumination and achieve color constancy. The first method uses denoising algorithms to smooth the images. We specifically study Gaussian filter, Median filter, Bilateral filter and Nonlocal-means filter for smoothing the images. We can also see a correspondence between the effectiveness of these denoising algorithms and estimation of the illuminant. The second method uses a modified version of cepstral analysis that has been very effective in speech processing. To estimate the illumination, we perform a frequency analysis of the image and extract the illumination, which is the low frequency component of the image and for that we have designed 2 low-pass filters – Butterworth and Chebyshev. In order to estimate the color of the illuminant, we then use the white-patch assumption [16].

2. PREVIOUS WORK

As the color of the image that we observe depends on the actual color of the light source and the camera property, the aim of the color constancy algorithms is to express:

$$I = \int I(\lambda)c(\lambda), \quad (1)$$

where $I(\lambda)$ is the color of the illuminant with wavelength λ and $c(\lambda)$ is the sensitivity of camera to λ .

To estimate the color of the illumination in a scene, many color constancy algorithms have been proposed. Some of the algorithms using low-level features for color constancy are the White-Patch algorithm [16], the Gray-World algorithm [5] and the Gray-Edge algorithm [21]. As shown in [21], all of these techniques can be combined together in one formula as follows:

$$\left(\int \left| \frac{\partial^n I^\sigma(x)}{\partial x^n} \right|^p dx \right)^{\frac{1}{p}} = k I^{n.p.\sigma}, \quad (2)$$

where n is the order of the derivative, p is the Minkowski norm and σ is the parameter for smoothing the image I .

Other more complicated techniques include gamut-based (GCIE) [11], color by correlation [12] or use neural networks [7]. Schaefer et al. [18] use a combined technique based on color by correlation and a dichromatic reflectance model. Chakrabarti et al. propose a technique that considers spatial dependencies between the pixels in the image to achieve color constancy [8].

3. OUR APPROACH

We present the following two methods to process the image and estimate the illumination before applying the White-Patch algorithm to get the color of the illumination – 1) Denoising techniques and 2) Cepstral analysis of the image.

3.1. Denoising Techniques

Denoising techniques are used in image processing to remove noise from the image by smoothing. The smooth image thus obtained is the illumination image. We study the following four methods:

3.1.1. Gaussian Filter

This technique is used to blur the image and therefore remove noise. Gijzen and Gevers have explored iterated local averaging [13] and can be considered similar to the Gaussian filter approach. The edge information is lost during Gaussian filter and this introduces error while estimating the illuminant.

3.1.2. Median Filter

The median filter chooses the median value of the neighborhood for a given pixel – every pixel has same number of color values above and below it in a neighborhood [20]. Smoothing using median filter may result in the loss of fine details of the image though boundaries may be preserved.

3.1.3. Bilateral Filter

In case of bilateral filter, every pixel of the image is replaced by the weighted sum of its neighbors [19]. The weights depend on two parameters – 1) Proximity of the neighbors to the current pixel (Closeness function) and 2) Similarity of the neighbors to the neighbors to the current pixel (Similarity function). The closer and the similar pixels are given higher weights. These two parameters can be combined to describe the bilateral filter as follows:

$$\mathbf{j}(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{i}(\eta) c(\eta, x) s(\mathbf{i}(\eta), \mathbf{i}(x)) d\eta, \quad (3)$$

where η is the neighboring pixel and k is the normalization term. The closeness function $c(\eta, x)$ and the similarity function $s(\mathbf{i}(\eta), \mathbf{i}(x))$ are:

$$c(\eta, x) = e^{-\frac{1}{2} \left(\frac{d(\eta, x)}{\sigma_d} \right)^2}, \quad (4)$$

$$s(\mathbf{i}(\eta), \mathbf{i}(x)) = e^{-\frac{1}{2} \left(\frac{\delta(\mathbf{i}(\eta), \mathbf{i}(x))}{\sigma_r} \right)^2}.$$

In case of the closeness function, $d(\eta, x) = \|\eta - x\|$ is the Euclidean distance between η and x whereas in case of the similarity function, $\delta(\mathbf{i}(\eta), \mathbf{i}(x))$ is the pixel value difference between η and x .

3.1.4. Non-Local Means Filter

The hypothesis behind non-local means (NL-means) technique is that for any image, the most similar pixels to a given pixel need not be close to it [4]. They could lie anywhere in the image. For comparing how similar the pixels are, instead of checking the difference between the pixel values (which is used in bilateral filtering), comparison of a window around the pixel is done.

The formulation of the NL-means filter is:

$$NLu(x) = \frac{1}{C(x)} \int e^{-\frac{(G_\rho * u(x+\cdot) - u(y+\cdot))^2(0)}{h^2}} u(y) dy, \quad (5)$$

where $u(x)$ is the observed intensity at pixel x , G_ρ is the Gaussian kernel with standard deviation ρ , h is the filtering parameter and $C(x)$ is the normalizing factor.

This means that image pixel $u(x)$ is replaced by a weighted average of $u(y)$ and this weight is given by the similarity between the Gaussian neighborhood of pixel x and pixel y .

To check for loss of details of the image after denoising, the noise is qualitatively visualized after applying the denoising algorithm and is called as method noise. Amongst the techniques mentioned here, NL-means filter performs best at preserving the original structure of the image [4].

3.2. Cepstral Analysis of the Image

The term ‘power cepstrum’ [3] of a signal as defined in [9] can be formulated as:

$$x_{pc}(nT) = (Z^{-1}(\log|X(z)|^2))^2 = \left\{ \frac{1}{2\pi j} \oint \log|X(z)|^2 z^{n-1} dz \right\}^2, \quad (6)$$

where $X(z)$ is the z -transform of the signal $x(nT)$.

A slightly different version of this technique is called homomorphic filter defined by Oppenheim and Schaffer [17], which does the inverse z -transform of the log of the magnitude of the z -transform of the signal. We follow a similar framework and do a Fourier analysis of the image and try to extract the illumination that is the low frequency component and use the following two low-pass filters:

3.2.1. Butterworth Filter

The Butterworth filter is used to get a flat response in the passband of the signal [6] and the ripple effect is negligible. Using higher order filters, we can get steeper roll-off. The response of an n -order Butterworth low pass filter is:

$$R(\omega) = \frac{1}{1 + \left(\frac{\omega}{\omega_c} \right)^{2n}}, \quad (7)$$

where ω is the frequency component, n is the order of the filter and ω_c is the cutoff frequency.

3.2.2. Chebyshev Filter

The Chebyshev filter has a steeper roll-off than Butterworth filter and there is more ripple in the passband. Using higher order filters, we can get steeper roll-off. The response of an n -order Chebyshev low pass filter is:

$$R(\omega) = \frac{1}{1 + \epsilon^2 C_n^2 \left(\frac{\omega}{\omega_c} \right)}, \quad (8)$$

where ϵ is the ripple, ω_c is the cutoff frequency and C_n is the n th order Chebyshev polynomial.

3.3. Estimation of the illuminant color

In order to estimate the color of the illumination, we find the maximum values of the red, green and blue channels across the entire illumination image. This is inspired from

the idea that a white patch in an image reflects the entire incident light, and thus can be used for estimating the illumination. Note that the maximum values of each channel need not be at the same pixel location of the image.

If C is the color of illumination, the corrected images as shown in Figure 1 can be obtained as follows:

$$\text{Color Corrected Image} = \frac{1}{K} \frac{\text{Original Image}}{C}, \quad (9)$$

where K is a constant.

4. EXPERIMENTS AND RESULTS

We evaluate our color constancy techniques on two widely used datasets. The first dataset is an indoor environment under varying illumination conditions and the illumination color is known [2]. There are 31 different scenes under 11 varying lighting conditions for a total of 321 usable images. The second dataset contains images from a real world environment that has around 11000 images [10]. The images are taken from 15 different scenes and randomly 10 images from each scene have been used for measuring the value of the illuminant. The annotated illuminant value is calculated from a gray ball that had been mounted on the camera while capturing the video but that has been excluded from estimating the illuminant value in our experiments.

In order to measure the error difference between the ground truth \mathbf{l}_{gt} and the estimated value of the illuminant \mathbf{l}_e , the angular difference ε between the two components can be calculated as:

$$\varepsilon = \cos^{-1}(\hat{\mathbf{l}}_{gt} \cdot \hat{\mathbf{l}}_e), \quad (10)$$

where $\hat{\mathbf{l}}_{gt} \cdot \hat{\mathbf{l}}_e$ is the dot product between normalized vectors. In order to measure the overall performance across the entire dataset, we use the median angular error [15].

The authors of [21] have provided an implementation of the existing color constancy algorithms using low-level features and the parameters were chosen from [21] that give the best results. The results from more complex color constancy algorithms like color by correlation, gamut mapping, neural networks etc. are reported in [1], [11] and [14]. GCIE algorithm using 11 lights perform very well because this technique uses the 11 illuminants that were used for testing as prior knowledge. The performance drops if more illuminants are used for training.

On a subset of the indoor environment shown in Table 1, Schaefer et al. [18] have reported a median error of **2.20°**. On the same subset of images, we get a median error of **1.89°** using the Chebyshev filter.

From Table 2, we can see that our methods have a **23%** improvement over the gray-edge algorithm. We also compared our algorithm with a very recent technique – ‘Beyond Bags of Pixels’ proposed by Chakrabarti et al. [8] and observed that our technique produces almost **26%** less error. The algorithms were implemented in MATLAB in Windows XP environment on a PC with Xeon processor and

the reported speed is for images with size 360 X 240 pixels. The speed of Bilateral and NL-means filter can be improved by using faster and optimized versions of those methods.

We can see that the performance of the denoising filters is better than the low-level techniques reported in Section 2 and comparable to the complex learning algorithms. Furthermore, the estimation of illuminant is even better using the cepstral analysis of the image.

Method	Parameters	Median
Gray-World	-	6.9
White-Patch	-	6.4
1 st order Gray-Edge	$p = 7$	3.2
2nd order Gray-Edge	$p = 7$	2.8
Color by Correlation	-	3.1
Gamut Mapping	-	2.9
Neural Networks	-	7.7
GCIE Version 3, 11 lights	-	1.3
GCIE Version 3, 87 lights	-	2.6

Gaussian filter	$\sigma = 5$	3.1
Bilateral filter	$n = 5, \sigma_d = 7, \sigma_r = 7$	3.0
Median filter	$n = 14$	2.6
NL-means filter	$w = 5, s = 7, h = 1.0$	2.7
Butterworth filter	$\omega_c = 0.09, o = 3$	2.4
Chebyshev filter	$\omega_c = 0.01, o = 3, \epsilon = 0.002$	2.5

Table 1: Median errors (in degrees) for the **indoor environment** along with the best parameter. n is the neighborhood for every pixel; w is the patch size for efficient computation and s is the local search area; o is the order of filter. The other parameters have been described in Sections 2 and 3.

Method	Parameters	Median
Gray-World	-	7.36
White-Patch	-	4.85
1st order Gray-Edge	$p = 6$	4.41
Beyond Bags of Pixels	-	4.58

Method	Parameters	Median	Speed
Gaussian	$\sigma = 5$	3.63	0.05
Bilateral	$n = 3, \sigma_d = 5, \sigma_r = 7$	3.40	2.16
Median	$n = 9$	3.42	0.39
NL-means	$w = 5, s = 7, h = 0.28$	3.39	39.10
Butterworth	$\omega_c = 0.09, o = 3$	3.48	0.18
Chebyshev	$\omega_c = 0.01, o = 2, \epsilon = 0.009$	3.39	0.17

Table 2: Median errors (in degrees) for the **real world environment** along with the best parameter and the average computational speed in seconds. n is the neighborhood for every pixel; w is the patch size for efficient computation and s is the local search area; o is the order of filter. The other parameters have been described in Sections 2 and 3.

5. CONCLUSIONS

In order to achieve color constancy, we have proposed two new techniques. The first method achieves color constancy

using denoising techniques viz., Gaussian filter, Median filter, Bilateral filter and Non-local means filter. While estimating illumination using the denoising techniques, we observed that the best results were obtained using a non-local means filter which best preserves the structure of the image. Therefore, intelligent smoothing helps to better estimate the illuminant value. The second method estimates the color of the illuminant in the frequency domain using cepstral analysis of the image and using a couple of low pass filters – Butterworth and Chebyshev filter. Experiments on two widely used datasets showed that our techniques give an improvement over existing state-of-the-art color constancy methods.

6. ACKNOWLEDGEMENT

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

7. REFERENCES

[1] K.Barnard, L.Martin, B.Funt and A.Coath. A comparison of computational color constancy algorithms-part ii:Experiments with image data. In *IEEE TIP*, Vol.11, No.9, 985-996, September 2002.

[2] K.Barnard, L.Martin, B.Funt and A.Coath. A dataset for color research. In *Color Research and Appln.*, Vol.27, No.3, 2002.

[3] B.Bogert, M.Healy and J.Tukey. The quefrequency alanysis of time series for echoes:cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. In *Proc. of the Symp.on Time Series Analysis*(M. Rosenblat,Ed),Chapter 15, 209-243, Wiley, 1963.

[4] A.Buades, B. Coll and J.M.Morel. Nonlocal image and movie denoising. In *IJCV*, Vol.76, No.2, 123-139, 2008.

[5] G.Buchsbaum. A spatial processor model for object color perception. *J. of the Franklin Institute*,310(1):1-26,1980.

[6] S. Butterworth. On the theory of filter amplifiers. *Wireless Engineer*,7:536-541,1930.

[7] V.Cardei, B.Funt and K.Barnard. Estimating the scene illuminant chromaticity by using a neural network. In *J. Opt. Soc. Am. A*. 19, 2374-2386, 2002.

[8] A.Chakrabarti, K.Hirakawa and T. Zickler. Color Constancy beyond Bags of Pixels. In *IEEE CVPR*, 2008.

[9] D.Childers, D.Skinner and R.Kemerait. The Cepstrum:A guide to processing. In *Proc. of IEEE*, Vol.65, No.10, Oct 1977.

[10] F.Ciurea and B.Funt. A large image database for color constancy research. In *Proc. of the 11th Color Imaging Conference*,160-164.IS&T – 2003.

[11] G.Finlayson and S.Hordley. Gamut constrained illumination estimation. In *IJCV*, Vol.67, No.1, 93-109, 2006.

[12] G.Finlayson, S.Hordley and P.Hubel. Color by correlation: A simple, unifying framework for color constancy. In *IEEE Trans. PAMI*, 23(11):1209-1221, November 2001.

[13] A.Gijsenij and T.Gevers. Color constancy by local averaging. In *ICIAPW*, September 2007.

[14] S.Hordley and G.Finlayson. Re-evaluating color constancy algorithm performance. In *Journal of the Optical Society of America*, Vol.23, No.5, 1008-1020, May 2006.

[15] S.Hordley and G.Finlayson. Re-evaluating color constancy algorithms. In *Proc. of the Int. Conf. on Pattern Recognition*, volume 1, 76-79, Washington DC, USA, 2004.

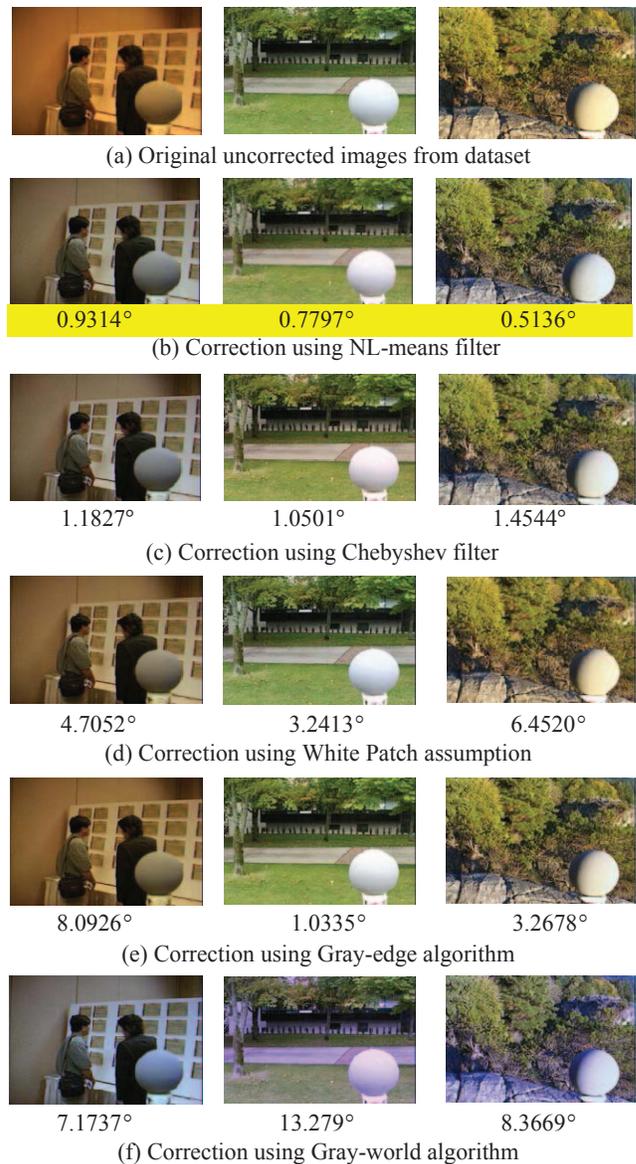


Figure 1: Examples of images from real world environment corrected after estimating the illumination by different methods along with their respective angular errors.

[16] E.Land. The retinex theory of color vision. *Scientific American*,237(6):108-128, December 1977.

[17] A.Oppenheim and R.Schafer. Homomorphic analysis of speech. In *IEEE Trans. On Audio and Electroacoustics*,Vol.16 No.2, 221-226, June 1968.

[18] G.Schaefer, S.Hordley and G.Finlayson. A combined physical and statistical approach to color constancy. In *IEEE CVPR* 2005.

[19] C. Tomasi and R.Manduchi. Bilateral filtering for gray and color images. In *6th Int. Conf. on Computer Vision*, 839-847, 1998.

[20] J.Tukey. *Exploratory data analysis*. Reading: Addison-Wesley, 1977.

[21] J.van de Weijer, T.Gevers and A. Gijsenij. Edge-based color constancy. In *IEEE TIP*, 16(9):2207-2214, 2007.