Reducing Worst-Case Illumination Estimates for Better Automatic White Balance

Milan Mosny and Brian Funt; School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada

Abstract

Automatic white balancing works quite well on average, but seriously fails some of the time. These failures lead to completely unacceptable images. Can the number, or severity, of these failures be reduced, perhaps at the expense of slightly poorer white balancing on average, with the overall goal being to increase the overall acceptability of a collection of images? Since the main source of error in automatic white balancing arises from misidentifying the overall scene illuminant, a new illumination-estimation algorithm is presented that minimizes the high percentile error of its estimates. The algorithm combines illumination estimates from standard existing algorithms and chromaticity gamut characteristics of the image as features in a feature space. Illuminant chromaticities are quantized into chromaticity bins. Given a test image of a real scene, its feature vector is computed, and for each chromaticity bin, the probability of the illuminant chromaticity falling into a chromaticity bin given the feature vector is estimated. The probability estimation is based on Loftsgaarden-Quesenberry multivariate density function estimation over the feature vectors derived from a set of synthetic training images. Once the probability distribution estimate for a given chromaticity channel is known, the smallest interval that is likely to contain the right answer with a desired probability (i.e., the smallest chromaticity interval whose sum of probabilities is greater or equal to the desired probability) is chosen. The point in the middle of that interval is then reported as the chromaticity of the illuminant. Testing on a dataset of real images shows that the error at the 90th and 98th percentile ranges can be reduced by roughly half, with minimal impact on the mean error.

Introduction

Illumination estimation is usually the first step in automatic white balancing of digital images. Once the RGB of the illuminant is known, it can be used to adjust the other image RGBs to make a more pleasing image. Many illumination-estimation algorithms have been proposed [2, 13, 15] and they tend to work reasonably well on average, but they often (perhaps, for 1 image out of 50) fail dramatically. The accuracy of illuminant estimates is often measured in terms of the angle in degrees between the estimated versus actual RGB of the illumination. Many methods are able to obtain mean and median angular errors under 4 degrees on the standard image test sets, but at the same time have maximum errors over 25 degrees. Images that are white balanced based on such inaccurate estimates are universally unacceptable. Although the mean and median measures are valuable error measures, users are perhaps more likely to be concerned with the number of failures than with the number of cases that are marginally poorer, say, having an error of 5 degrees instead of 4. We do not evaluate user preference metrics here, but presuppose that the failure cases are important, and propose a way of using the results from existing illumination-estimation algorithms that combines them in a probabilistic way that leads to lower large errors.

There are two general approaches to computational color constancy. The goal of the first approach is to create illuminant-invariant descriptors. Examples include [7, 10, 12]. The second approach aims to predict what the image of the scene would be under a canonical illuminant. This is usually accomplished by estimating the RGB of the scene illumination, and then adjusting the image accordingly [2, 13]. This paper focuses on the illumination-estimation step of this second approach.

State-of-the-art illumination-estimation algorithms [13] perform well in terms of median error (i.e., the error at the 50th percentile). However, little attention has been given to reducing the errors at the 90th or 99th percentiles. If at least 99% of our images are required to be acceptable then it is the error at the 99th percentile that is important.

This paper presents a new method called the Reduced Worst Case algorithm (RWC) for minimizing the errors at any specified percentile. For example, it is possible to tune the algorithm to minimize the 90th or 99th percentile errors. The algorithm combines image features consisting of the output from an existing algorithm such as Edge-based Color Constancy [22] or MaxRGB [9] with a characterization of the image gamut in terms of its minimum rgb (r=R/(R+G+B) etc.) and maximum rgb values [18]. The RWC algorithm is evaluated on the standard SFU database of 321 images [3]. The results show a significant decrease in the high percentile error accompanied by only a modest increase in the median error.

The paper is structured as follows: Section 2 describes related work. Section 3 describes the algorithm and features used in more detail. Section 4 presents the results of the accuracy tests for various combinations of features and desired percentile settings. Section 5 is the conclusion.

Related Work

There are illumination-estimation algorithms [5, 4] that combine multiple clues and/or results from different algorithms using an average or weighted average. Various forms of consensus were explored by Bianco *et al.* [4]. Machine learning techniques for combining results were explored by Li *et al.* [16] and Cardei *et al.* [5]. However, none of these methods attempts to minimize worst-case errors at a given percentile.

Perhaps closest to the RWC algorithm proposed here is that of Chakrabarti et al. [6], which uses a multivariate Gaussian probability distribution of features derived from the spatial frequencies of training images of scenes illuminated by a canonical illuminant. As well, there is the algorithm of Schaefer et al. [21]

that combines algorithms capable of producing the likelihood of an input image being illuminated by a particular illuminant.

The algorithm we propose for combining the features differs from that of Chakrabarti et al. in that a Gaussian (and therefore symmetric) distribution of the feature probabilities is not assumed. Moreover, the algorithm is trained using scenes illuminated by wide set of illuminant spectra and therefore is not limited to von-Kries diagonal transforms when constructing the models. Unlike Schaefer's algorithm, RWC can combine outputs of any algorithm regardless of whether or not they produce likelihood estimates. Neither Chakrabarti's nor Schaefer's algorithms focus on minimizing high-percentile errors as proposed in this paper. In terms of minimizing large errors, Cubical Gamut Mapping [18] attempts to minimize the maximal error, but it cannot be tuned to minimize the error for a given percentile.

Proposed Reduced Worst Case Algorithm

The proposed RWC algorithm estimates each of the three channels of the rgb chromaticity separately. Although any one of the channels in principle can be calculated from the other two, all three are estimated separately in order to improve the stability of the final estimate. Values in each chromaticity channel are quantized into N bins of equal size. Given an input image, the algorithm estimates the probability of the scene illuminant having the chromaticity associated with a particular bin. Once the probability distribution estimate for a given chromaticity channel is known, the smallest interval that is likely to contain the right answer with a desired probability (i.e., the smallest chromaticity interval whose sum of probabilities is greater or equal to the desired probability) is chosen. The point in the middle of that interval is then reported as the chromaticity of the illuminant. Given an accurate probability distribution, the algorithm will not make an error greater than ½ of the interval in the desired percentage (e.g., 90%) of the images. Figure 1 shows an example of the computed probability distribution estimate for a single channel, an interval covering chromaticity bins with at least 90% chance, and the resulting chromaticity returned by the algorithm. Note that the answer does not have to correspond to the maximum likelihood estimate.

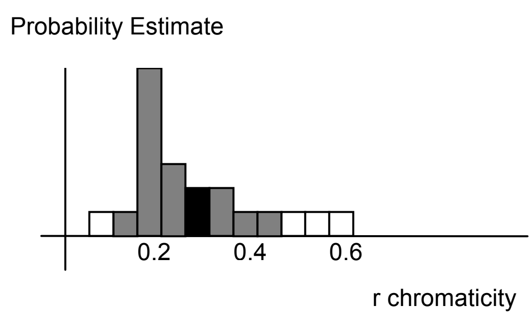

Figure 1. *An example of the probability estimate of the illuminant's r chromaticity. The smallest interval covering the desired percentile is shown in grey. The middle of the interval is shown in solid black. The algorithm returns the r chromaticity of 0.3, even though the maximum likelihood estimate for the r chromaticity is 0.2.*

The algorithm depends on correctly estimating the probability distribution of the illuminant chromaticity as a function of the chromaticity component bins. For each bin c_i, a model is created consisting of all feature vectors, F_{ij}, collected from training images Tij, $j=1, \ldots M_i$ whose actual illuminant corresponds to the chromaticity component bin c_i represents.

The feature vector F_{ij} contains:

- The estimated illumination chromaticity rgb for image T_{ij} provided by each of the underlying algorithms (e.g., from MaxRGB, Greyworld, and Edge-based Color Constancy),
- The minimum r, minimum g, and minimum b from T_{ij},
- The maximum r, maximum g, and maximum b from T_{ij}.

The minimum rgb and maximum rgb together provide a rough measure of the image's color gamut. As shown by Forsyth [8], the gamut is a useful feature because the illumination directly affects the gamut of image colors.

Given an input image, I, of a scene taken under an unknown illuminant, its feature vector, F, is constructed the same way by concatenating the chromaticity estimates of the underlying algorithms along with the minimum and maximum rgb values. To estimate the probability $P(c_i|F)$ that the unknown illuminant's chromaticity corresponds to bin c_i, Bayes rule is used:

$$P(c_i \mid F) \sim P(F \mid c_i)\, P(c_i) \tag{1}$$

The probability $P(F|c_i)$ of feature vector F belonging to bin c_i is based on Loftsgaarden-Quesenberry multivariate density function estimation [17] for the point in feature space occupied by F relative to all the training feature vectors F_{ij} in the bin c_i. The value $f_i(F)$ of the probability density function f_i at point F is estimated as

$$f_{i\,est} = \frac{k(n_i)-1}{n_i} \cdot \frac{\Gamma(d/2)\,d}{2r^d \pi^{d/2}} \tag{2}$$

where n_i is number of training points in the bin c_i, $k(n)$ is a non-decreasing sequence of positive integers (smallest integer greater or equal to $n^{1/4}$ is used here), d is the dimensionality of F, r is a radius of the smallest sphere centered around F that covers at least $k(n_i)$ training points $F_{i,j}$ from bin c_i, and Γ is the gamma function. The term $(\Gamma(d/2)d)/(2r^d\pi^{d/2})$ is an inverse of the volume of a sphere of radius r in a d-dimensional space. Assuming that $P_{est}(F|c_i) \sim f_{i\,est}(F)$ and removing the constant terms yields

$$P_{est}(c_i \mid F) \sim \frac{k(n_i)-1}{r^d}$$

As a result, the probability estimate $P_{est}(c_i|F)$ in Eq. (1) is easily computed from the feature vectors $F_{i,j}$ obtained during training.

Results

Simon Fraser University dataset [3] includes reflectance spectra, illuminant spectra, a set of test images with measured illumination and associated camera spectral sensitivity functions. The 1995 reflectance spectra are of Macbeth colorchecker patches, Munsell chips, DuPont paint chips, and other objects and surfaces. The dataset includes a set of 87 spectra that uniformly cover a convex region of rg-chromaticity space containing real measured illuminants. The camera sensor sensitivities are for the SONY DXC-930 camera used to capture the test images.

For training, approximately 6,500 synthetic images were generated [18] using the 1995 reflectance spectra, the set of 87 illuminant spectra, and the camera sensor sensitivities. The images were generated using a random illuminant and between 2 and 32 random reflectance spectra. Each scene contained two perpendicular planes forming a background and randomly placed spheres of random size. A computer graphics image of the scene was then generated using PovRay 3.6 ray tracing graphics software [19]. Sample images are shown in Figure 2.

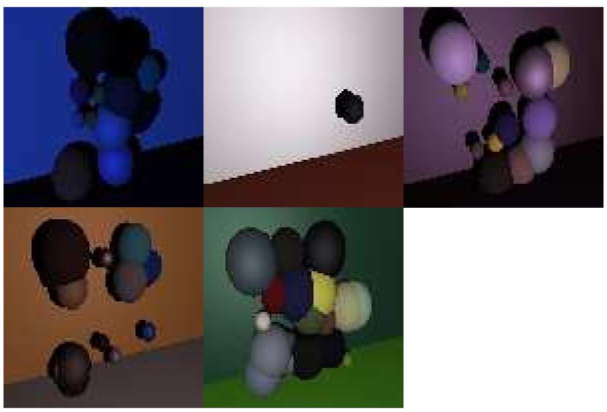

Figure 2. *Five examples of synthetic training images generated using POWRay.*

All testing is based on captured, not synthesized, images. The SFU color dataset [3] includes images of 51 scenes illuminated by 11 different illuminants. The images are provided in 16-bit linear RGB triplets in the color space of the SONY DXC-930 video camera. The dataset is divided into four sets of 223, 98, 149 and 59 images, respectively. The set of 223 images contains minimal specularities, the set of 98 images contains non-negligible dielectric specularities, the set of 149 images contains metallic specularities, and finally the set of 59 images contains fluorescent surfaces. In order to be able to compare the performance of RWC to that of others as reported in the literature, the algorithm is evaluated on the 321 images from the combined non-specular and dielectric sets. Unfortunately, the other publicly available test sets are not useful for testing RWC because they do not include the camera sensor sensitivities (e.g., the Gehler set [11]) or they are all of outdoor scenes (e.g., the Barcelona set [20]) and hence include only a very limited range of illuminants.

Features used in the algorithms include illumination estimates from MaxRGB and Edge-based Color Constancy [22] and the minimum and maximum rgb chromaticities in the image. The following combinations of features were evaluated:

1. MaxRGB
2. MaxRGB plus Edge-based Color Constancy
3. MaxRGB plus minimum rgb and maximum rgb
4. Edge-based Color Constancy plus MaxRGB, minimum rgb, and maximum rgb.

The following types of pre-processing were applied to the images depending on the method being used:

- Dark-pixel removal. Dark pixels, that is, those with $(R + G + B) < Threshold$, are removed. The *Threshold* is set relative to the average of the $R+G+B$ values collected from all pixels in the image.
- Clipped-pixel removal. Pixels whose RGB values are above the upper limit of the camera's dynamic range are removed. The threshold is set to 98% of the maximum RGB value.
- Gaussian smoothing. Gaussian smoothing reduces the noise and is used as a pre-processing step for the Edge-based Color Constancy method [22, 14].
- Even blocks pre-processing [18]. The size of the neighborhood N is set to 5.

MaxRGB and the computation of the minimum rgb and maximum rgb features are all subjected to dark-pixel removal, clipped-pixel removal and even-blocks pre-processing. Edge-based Color Constancy is subjected to Gaussian smoothing, which is an intrinsic part of that algorithm.

The results are evaluated in terms of angular error [1, 2, 13, 15]. Treating rgb chromaticities as vectors in a 3D space, the angular error is the angle in degrees between the rgb chromaticity of the estimated illuminant and the rgb chromaticity of the actual scene illuminant. Figures 3 and 4 and Table 1 show the results.

The plot in Figure 3 shows the angular errors obtained using MaxRGB, Edge-based Color Constancy, and 4 variants of the RWC algorithm corresponding to the 4 different feature sets. The variants are all set to minimize the 90^{th} percentile error. From Figure 3, it is easy to see that RWC outperforms both MaxRGB and Edge-based Color Constancy for 75^{th} percentile and higher errors. At the 90^{th} percentile, the difference is quite significant, with the error reduced by almost half. The variants that include the minimum rgb and maximum rgb features obtain errors of less than 10 degrees for up to 93% of the test set, while at the same time performing similarly to Edge-based Color Constancy in the lower portion of the percentile range. RWC running with MaxRGB and Edge-based features outperforms the Edge-based Color Constancy algorithm on the whole range. It is interesting to see that MaxRGB performs extremely well for the first 40% of the percentile range.

The plot in Figure 4 shows the effect of choosing different percentiles at which to minimize the illumination-estimation error. The figure compares MaxRGB alone to RWC using the estimates from MaxRGB and Edge-based Color Constancy as features when minimizing the 50^{th}, 90^{th}, 98^{th} and 99.9^{th} percentile errors. The 50^{th}-percentile-optimized algorithm outperforms the others in the 40-65 percentile range, the 90^{th}-percentile-optimized leads in the 65-83 percentile range, the 98^{th}-percentile-optimized algorithm leads in 84-90 percentile range and the 99.9^{th}-percentile-optimized algorithm leads at the high-percentile errors. The differences between the desired best performance and actual best performance percentile range are likely caused by imperfections in the probability estimation.

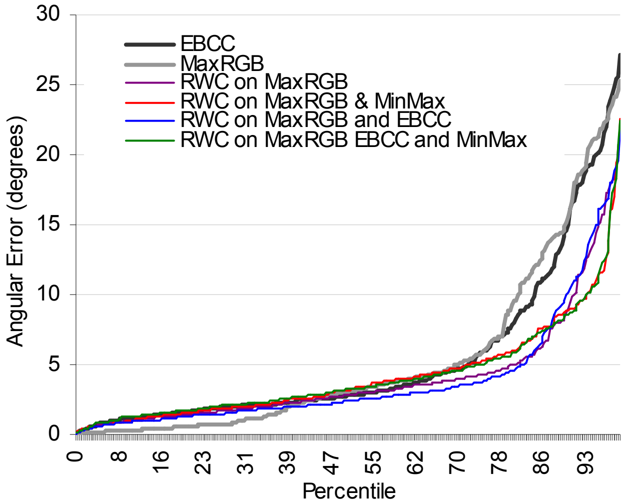

Figure 3. *Performance of MaxRGB algorithm, Edge-based Color Constancy (EBCC) and 4 variants of the proposed RWC algorithm corresponding to the 4 different feature sets described above. The algorithms attempt to minimize the 90th percentile error. The x-axis ranges from 0 to 100%, and the y-axis shows angular error at the given percentile.*

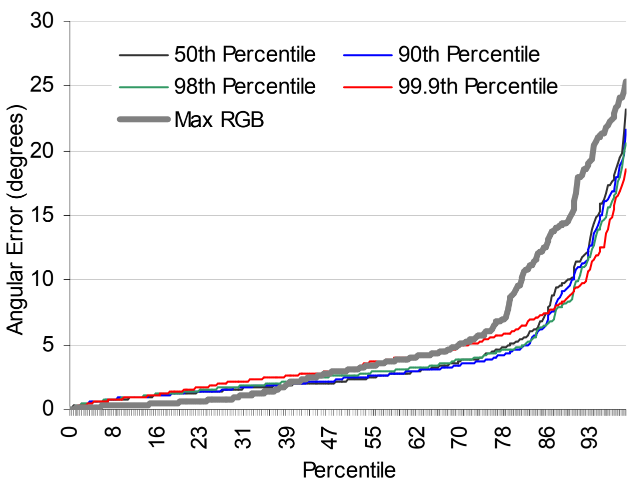

Figure 4*. Performance of MaxRGB versus the RWC algorithm with features of MaxRGB and Edge-based Color Constancy minimizing the 50th, 90th, 98th and 99.9th percentile errors.*

Table 1 shows the performance of the various algorithms when RWC optimizes the 90th percentile angular error. The comparison is in terms of the median error, the error at the 90th and 98th percentiles, the maximum error, the mean error and the root mean square error. The RWC variant based on MaxRGB combined with Edge-based Color Constancy outperforms either of them taken separately, especially at the 90th percentile. Adding the image gamut information provided by of the minimum rgb and maximum rgb yields a further improvement at the 98th percentile.

Table 1: Performance of the Edge-based Color Constancy (EBCC), MaxRGB and 4 variants of RWC (using MaxRGB, using both MaxRGB and EBCC, using MaxRGB with the minimum and maximum rgb, using MaxRGB, EBCC and the minimum and maximum rgb) when optimizing the 90th percentile angular error reported in terms of the median error, root mean square error, mean error, 90th percentile error, 98th percentile error, and maximum error.

Method	Avg	RMS	50th	90th	98th	Max
EBCC	5.3	7.8	2.8	14.4	23.4	27.2
MaxRGB	5.2	8.2	3.0	14.9	22.8	25.3
RWC on MaxRGB	4.0	5.6	2.8	**8.3**	17.4	**21.5**
RWC on MaxRGB & EBCC	**3.8**	5.6	**2.4**	9.6	16.9	21.7
RWC on MaxRGB & minmax	4.1	5.4	3.2	8.7	13.9	22.6
RWC on MaxRGB, EBCC & minmax	4.2	**5.3**	3.2	8.5	**13.8**	22.5

Conclusion

Our goal was to improve estimation of the scene illuminant's chromaticity in the sense of reducing the number and seriousness of poor estimates even if that reduction comes at the cost of slightly poorer estimates on average. A novel algorithm was presented that accomplishes this goal, in essence by hedging the bets, so to speak, of estimates obtained from other standard illumination-estimation algorithms. The RWC is a general framework so other algorithms can be substituted for the MaxRGB and Edge-based algorithms tested thus far. Testing on a dataset of real images shows that the proposed reduced-worst-case method can reduce the error at the 90th or 98th percentile range by as much as 50% with only a marginal increase in the mean error.

Acknowledgement

The financial support of the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

[1] K. Barnard, V. Cardei, and B.V. Funt. A comparison of computational color constancy algorithms-part i: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, 11(9):972–984, September 2002.

[2] K. Barnard, L. Martin, A. Coath, and B.V. Funt. A comparison of computational color constancy algorithms-part ii: Experiments with image data. *IEEE Transactions on Image Processing*, 11(9):985–996, September 2002.

[3] K. Barnard, L. Martin, B.V. Funt, and A. Coath. A data set for colour research. *Color Research and Application*, 27:147–151, 2002.

[4] S. Bianco, F. Gasparini, and R. Schettini. Consensus-based framework for illuminant chromaticity estimation. *Journal of Electronic Imaging*, 17(2):023013, 2008.

[5] Vlad C. Cardei and Brian V. Funt. Committee-based color constancy. In *The Seventh Color Imaging Conference: Color Science, Systems, and Applications: Putting It All Together, November 16-19, 1999, Scottsdale, Arizona, USA*, pages 311–313. IS&T - The Society for Imaging Science and Technology, 1999.

[6] Ayan Chakrabarti, Keigo Hirakawa, and Todd Zickler. Color constancy with spatio-spectral statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[7] Graham D. Finlayson, Steven D. Hordley, John A. Marchant, and Christine M. Onyango. Colour invariance at a pixel. In Majid Mirmehdi and Barry T. Thomas, editors, *BMVC*. British Machine Vision Association, 2000.

[8] D.A. Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 5(1):5–36, August 1990.

[9] Brian Funt and Lilong Shi. The rehabilitation of maxrgb. In *Proceedings of IS&T Eighteenth Color Imaging Conference, San Antonio*, Nov. 2010.

[10] B.V. Funt and G.D. Finlayson. Color constant color indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):522–529, May 1995.

[11] Peter V. Gehler, Carsten Rother, Andrew Blake, Thomas P. Minka, and Toby Sharp. Bayesian color constancy revisited. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[12] J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, December 2001.

[13] A. Gijsenij, T. Gevers, and J. van de Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, (in press), 2011.

[14] Arjan Gijsenij and Theo Gevers. Color constancy by local averaging. In *2007 Computational Color Imaging Workshop (CCIW'07), in conjunction with ICIAP'07, Modena, Italy*, pages 1–4, 2007.

[15] S.D. Hordley and G.D. Finlayson. Reevaluation of color constancy algorithm performance. *Journal of the Optical Society of America*, 23(5):1008–1020, May 2006.

[16] Bing Li, Weihua Xiong, De Xu, and Hong Bao. A supervised combination strategy for illumination chromaticity estimation. *ACM Trans. Appl. Percept.*, 8(1):5:1–5:17, November 2010.

[17] D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

[18] Milan Mosny and Brian Funt. Cubical gamut mapping colour constancy. In *Proceedings of CGIV2010 IS&T Fifth European Conf. on Colour in Graphics, Imaging and Vision, Joensuu*, June 2010.

[19] Persistence of Vision Pty. Ltd. Persistence of vision raytracer (version 3.6), 2004. computer software, downloaded in 2009.

[20] C. A. Párraga, J. Vazquez-Corral, and M. Vanrell. A new cone activation-based natural images dataset. *Perception*, 36:180, 2009.

[21] Gerald Schaefer, Steven D. Hordley, and Graham D. Finlayson. A combined physical and statistical approach to colour constancy. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 148–153. IEEE Computer Society, 2005.

[22] J. van de Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214, September 2007.