

Automatic White Balancing via Gray Surface Identification

Weihua Xiong¹, Brian Funt^{**}, Lilong Shi^{**}, Sung-Su Kim^{***}, Byoung-Ho Kang^{***}, Sung-Duk Lee^{***} and Chang-Yeong Kim^{***}

¹ Omnivision Corporation, CA, USA¹

^{**} Simon Fraser University, Burnaby, Canada

^{***} Samsung Advanced Institute of Technology, Korea

Abstract

The key to automatic white balancing of digital imagery is to estimate accurately the color of the overall scene illumination. Many methods for estimating the illumination's color have been proposed [1-6]. Although not the most accurate, one of the simplest and quite widely used methods is the gray world algorithm [6]. Borrowing on some of the strengths and simplicity of the gray world algorithm, we introduce a modification of it that significantly improves on its performance while adding little to its complexity.

Introduction

The key to automatic white balancing of digital imagery is to estimate accurately the color of the overall scene illumination. Many methods for estimating the illumination's color have been proposed [1-6]. Although not the most accurate, one of the simplest and quite widely used methods is the gray world algorithm [6]. Borrowing on some of the strengths and simplicity of the gray world algorithm, we introduce a modification of it that significantly improves on its performance while adding little to its complexity.

The standard gray world algorithm is based on the assumption that the average surface color in a scene is gray so that when an image's colors are averaged, any departure from gray reflects the color of the scene illumination. The proposed extension first identifies colors that are likely to be from truly gray surfaces, and then averages only those colors. The trick is in the identification of gray surfaces. Note that we must make a distinction between the color of the surface as it would appear under white light and the image color of that same surface under the unknown scene illumination. We can not simply average image colors that are gray since that would tell us nothing other than that gray colors are gray. To find the surfaces that are gray, but do not necessarily appear gray in the image because of the effect of the illumination, we use a color coordinate system [7] that encodes illumination and surface reflectance along different axes. By comparison, Cooper [8] describes how to determine near neutral regions based on image segmentation in $L^*a^*b^*$ coordinates.

LIS Color Coordinates

The goal of the LIS color coordinate system is to represent the 3 components of a color in terms of the underlying physical components that generated the color, in particular, luminance/intensity, incident illumination color, and the surface reflectance color. Of course, this goal cannot actually be met without additional information, but it can be approximated to a useful extent. Since the coordinates represent luminance, illumination color and surface reflectance as separate dimensions, hence the designation "LIS coordinates". The LIS coordinate system was proposed Finlayson and Hordley [7], although they do not call it by that name.

Experimenting with the LIS channels showed that points in it having an S coordinate of zero were generally gray. They are not just gray in RGB image space, but represent gray surface colors because they are in the reflectance space. To the extent that the S coordinate actually does represent reflectance and truly is independent of the illumination, this means that we can identify gray surfaces in an image independent of whether or not they have $R=G=B$. The strategy for the proposed new automatic white balance (AWB) method, therefore, is to use the LIS coordinates to identify gray surfaces in the image, and then use these grays to estimate the illuminant color. For this final step, we convert back to the original RGB color space of the image and average the chromaticities of the grays. We call this method GSI (gray surface identification).

Figure 1 shows an example of the gray-pixel detection results. The detected pixels are marked in white in 1(b). The chromaticities of these pixels in 1(a) can be averaged to obtain the color of the illuminant for AWB.

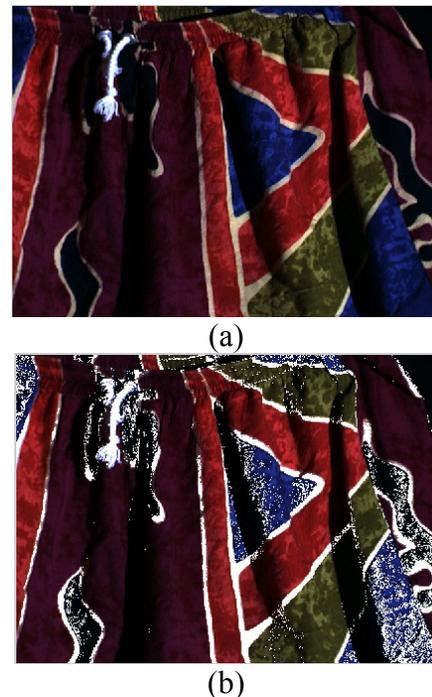


Figure 1 (a) Input image; (b) Pixels identified as gray are indicated in white.

LIS coordinates [7] are based on two assumptions: (1) that the illuminants are blackbody radiators; and (2) that the camera's response functions are narrowband and can be modeled as Dirac delta functions. The implication of the first assumption is that the illuminants can be modeled as a function of a single parameter, namely, the blackbody temperature. The implication of the second assumption is that each of the RGB channels is affected

¹ This work was done while the author was a PH.D. student at Simon Fraser University.

by only a single distinct wavelength of the incoming spectrum. Under these assumptions, Finlayson and Hordley [7] show that for a given camera, $(\log R/G, \log B/G)$ is an illumination-invariant color chromaticity space in which the values of same surface under various illuminations tend to fall on a straight line and lines from different surfaces are parallel. For a fixed surface reflectance, varying the intensity and color temperature of the illumination incident on it causes the logarithm of the camera response $[\log R, \log G, \log B]$ to move within a plane. Different surface reflectances yield parallel planes. The S axis of the LIS system is defined perpendicular to these planes. The L (luminance) axis and I (illumination ‘color’) axis are then orthogonal to the S axis.

Although in theory the logarithm of the camera responses $[\log R, \log G, \log B]$ obtained from a given surface under all possible colors and intensities of illumination are predicted to lie in a plane, do they in practice? Clearly, the blackbody-radiator and Dirac-delta assumptions are strong ones, and are likely to be violated. However, for $[\log R, \log G, \log B]$ data synthesized based on the SONY DXC-930 sensitivity functions, the 102 illuminant spectra from the Simon Fraser University database [9] and the surface reflectance of the 24 Macbeth Colorchecker surface patches, PCA (principal component analysis) determines the plane and establishes that the first 2 dimensions explain 99.1% percent of the variance. These 102 illuminants are not specifically blackbody radiators, but common light sources found around a university campus. Similarly, although the camera sensitivity functions [9] are relatively sharp with little overlap between them, they are taken from a real camera, and certainly violate the Dirac-delta assumption. Despite violating the assumptions, the fit of a plane to the data is surprisingly good.

GSI Implementation

The first issue in terms of implementing the GSI color constancy algorithm is that the LIS system is camera dependant and must be determined for the camera being used. There are two methods to do this depending on whether or not the camera’s spectral sensitivity response functions are known. If they are known, then they can be used to calculate camera responses for spectra synthesized as the product of illuminant and reflectance spectra chosen from a database of spectra. If the camera’s spectral sensitivity curves are not known, then real values can be obtained by using the camera to take images of a gray card under several different illuminants. PCA is then applied to the logarithm of RGBs from the gray card. The vector corresponding to the maximal eigenvalue forms the intensity axis, the next vector forms the illumination axes, and the vector corresponding to the least eigenvalue is the surface reflectance axis.

To estimate the illumination for an image of N pixels $[R_i, G_i, B_i]$, each pixel is first classified as to whether or not it belongs to the class of gray pixels. To classify a pixel, the logarithm of each channel is taken producing $[\log R_i, \log G_i, \log B_i]$ which is then projected onto the S axis of the LIS coordinate system via vector inner product. If the resulting value is less than a specified threshold value then the pixel is classified as gray.

The GSI method estimates the color $[R_e, G_e, B_e]$ of an image’s illumination according to

$$\begin{cases} R_e = \frac{1}{N} \sum_i w_i R_i \\ G_e = \frac{1}{N} \sum_i w_i G_i \quad \text{if } \text{isgray}([R_i, G_i, B_i]) \text{ then } w_i = 1 \text{ else } w_i = 0 \\ B_e = \frac{1}{N} \sum_i w_i B_i \end{cases}$$

where ‘isgray’ is the test that classifies pixels as gray or not.

An example of the GSI method is shown in Figure 1. The *isgray* test identifies as gray those pixels from Figure 1(a) that are shown in white in Figure 1(b)). The true scene illumination as measured from a gray card is $[0.2476, 0.2910, 0.4614]$. The standard gray world method averages the RGBs of all pixels so that the estimated illumination is found to be $[0.4748, 0.2348, 0.2903]$. The GSI method, however, averages only the RGB of pixels that pass the *isgray* test with the result that the illumination is estimated to be $[0.2810, 0.3290, 0.3899]$. Clearly, this latter estimate is much closer to the true value. This example shows the potential of the GSI method; rigorous tests are presented in the next section.

Experiments

The GSI method was implemented in MATLAB 7.0.1 [10] To evaluate GSI’s illumination estimation and compare it to other methods, the algorithm was tested on two datasets of real images. The first one includes the 321 images of the SFU dataset [9], which are of scenes in a laboratory setting. The second set is the much larger and more varied image collection that Ciurea et. al. [11] built using a digital video camera.

In evaluating performance, we use error measures based on Euclidean distance and angular difference between the estimated and true illumination chromaticity values. Given an illumination estimate $[R_e, G_e, B_e]$, its corresponding chromaticity values are

$$r_e = \frac{R_e}{(R_e + G_e + B_e)}, g_e = \frac{G_e}{(R_e + G_e + B_e)}, b_e = \frac{B_e}{(R_e + G_e + B_e)}$$

Let $[r_r, g_r, b_r]$ be the true illumination chromaticity. The distance error in 2D chromaticity space and angular error in 3D chromaticity space are defined as:

$$E_{i-dist} = \sqrt{(r_r - r_e)^2 + (g_r - g_e)^2}$$

$$E_{i-angular} = \cos^{-1} \left[\frac{(r_r, g_r, b_r) \circ (r_e, g_e, b_e)}{\sqrt{r_r^2 + g_r^2 + b_r^2} \times \sqrt{r_e^2 + g_e^2 + b_e^2}} \right] \times \frac{2\pi}{360}$$

For test set of N images, we also report the maximum, median and RMS errors

$$RMS_{dist} = \frac{1}{N} \sqrt{\sum_{i=1}^N E_{i-dist}^2}$$

$$RMS_{angular} = \frac{1}{N} \sqrt{\sum_{i=1}^N E_{i-angular}^2}$$

To evaluate whether one method is statistically better than another, we also use the Wilcoxon signed-rank test with 0.01 threshold for accepting or rejecting null hypothesis [12].

The first experiment uses Barnard’s [9] 321 images captured using a calibrated SONY DXC-930 camera. These images are from 33 different scenes under 11 different lights that represent a cross-section of common lights. Since the spectral sensitivity functions of the camera are known and the calibration images are available on the Internet [9], this data set provides a means of comparing LIS coordinates extracted based on synthetic versus real data. For the synthetic case, we synthesize RGB values for the measured percent spectral reflectance of 24 Macbeth ColorChecker

patches and the spectral power distributions of 102 illuminants at 15 different intensities values. Applying PCA to this data, we find the LIS axes as row vectors:

[0.5994 0.5871 0.5441],
 [0.6421 0.0482 -0.7651],
 [0.4729 -0.8132 0.3358].

To compute the LIS coordinates from real data, we have the RGB values from the gray card under the 11 different illuminants. These RGBs are then scaled by 15 different factors to create a RGBs that vary in intensity. PCA is applied to the logarithms of the resulting 165 RGBs. The LIS axes obtained are:

[0.6040 0.5807 0.5459],
 [0.6429 0.0499 -0.7643],
 [0.4711 -0.8126 0.3432].

Clearly, the two methods produce very similar results. The advantage of the real data method is that it is generally easier to collect images of a gray card under a dozen or so different illuminants than it is to determine a camera's spectral sensitivity functions.

Having determined the LIS coordinates, we can proceed to test the GSI method. Since 321 is a small number of images, we use leave-one-out cross-validation [13] in evaluating its performance and that of competing methods. Each method is trained on 320 of the images and tested on the one remaining image. This procedure is repeated a total of 321 times so that each image can be tested. In the case of GSI, the training consists of choosing the optimal *isgray* threshold minimizing the median angular error over the training set. Table 1 compares GSI performance to that of Support Vector Regression [1] both on RGB data (3D) and chromaticity data (2D), to Shades of Grey [4] with the optimal choice of norm, to Max RGB [5] which takes the maximum in each of the 3 color channels as the illumination color, and to standard Grayworld [6].

Our second experiment is based on the Ciurea et. al.[10] dataset. Each image contains a matte gray ball in its lower right hand corner. The average chromaticity value of the pixels in the brightest region of the grayball is used as a measure of the color of the scene illumination in camera coordinates. The camera was uncalibrated, so we used the real data method to calculate the LIS coordinates for it based on RGBs from the gray ball.

The original image database includes 11,346 images. However, many of these images have very good color balance (i.e., RGB of the gray ball is gray) which could bias the testing of the illumination estimation methods. Therefore, we eliminated from the data set the majority of the correctly balanced images so that the overall distribution of the illumination color is more uniform, as can be seen in Figure 2. The resulting data set contains 7661 images.

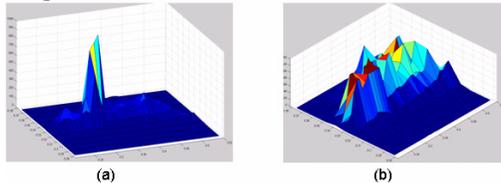


Figure 2 (a) The original data set contains 11,346 images, but the illumination chromaticities cluster around gray (0.33, 0.33). (b) The reduced data set contains 7661 images with a more uniform distribution of illumination chromaticity.

As shown in Figure 3, the images are cropped to remove the gray ball, which is located at a fixed location in the lower right quadrant. The resulting image size is 240 by 240.



(a)



(b)

Figure 3 (a) Original image containing gray ball from which the color of the scene illumination is determined. (b) Cropped image to be used for algorithm testing with gray ball removed.

Neighboring images in the database tend to be related to one another, so we partitioned it into two disjoint subsets based on the geographical location where the images were acquired. Subset A contains 3581 images, and subset B 4080. First, subset A is used for training and subset B for testing, then vice versa. The errors from both tests are combined in the entries in Table 3 and Table 4.

Conclusion

A new color constancy method, GSI, is proposed that is based on detecting pixels corresponding to gray surface reflectance—which is not necessarily the same as gray image color—and using their average image color as an indicator of the color of the overall scene illumination. The gray surfaces are found by first transforming the image RGB values to the LIS coordinate system with axes that roughly correspond to luminance, illumination ‘color’ and reflectance. In LIS coordinates, values of *S* near zero tend to be gray. Tests on real images show the GSI method works better than Shades of Gray, Grayworld and Max RGB. While it is not quite as accurate as 3D SVR, it is much faster, does not require training, and is substantially simpler to implement.

Acknowledgements

This work was sponsored by the Samsung Advanced Institute of Technology.

References

- [1] B.V. Funt, and W.H. Xiong. “Estimating Illumination Chromaticity via Support Vector Regression,” *Proceedings of 12th Color Imaging Conference*, pp.47-52, (2004)
- [2] V. Cardei, B. Funt, and K. Barnard, “Estimating the Scene

Illumination Chromaticity Using a Neural Network,” *Journal of the Optical Society of America A*, Vol. 19, No. 12, pp. 2374-2386, (2002)

[3] G. D. Finlayson, S. D. Hordley and P. M. 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, (2001)

[4] G. D. Finlayson and E. Trezzi, “Shades of Gray and Colour Constancy,” *Proc. IS&T/SID Twelfth Color Imaging Conference: Color Science, Systems and Applications, Society for Imaging Science and Technology*, Scottsdale, pp. 37-41, (2004)

[5] G.D. Finlayson. “Retinex viewed as a gamut mapping theory of color constancy,” *Proc. AIC International Color Association Color 97*, Vol. 2, pp. 527-530, (1997)

[6] B. Buchsbaum. “A spatial processor model for object color perception,” *Journal of the Franklin Institute*, Vol. 31, pp. 1-26, (1980)

[7] G.D. Finlayson, and S.D. Hordley, “Color Constancy at a pixel,” *Journal of the Optical Society of America A*, Vol. 18, No. 2, pp. 253-264, (2001)

[8] Ted Cooper, “Color Segmentation as an Aid to White Balancing for Digital Still Cameras,” *Proc. SPIE*, Vol. 4300, pp. 164-171, (2001)

[9] K. Barnard, L. Martin, L., B.V. Funt, and A. Coath, “A Data Set for Color Research,” *Color Research and Application*, Vol. 27, No. 3, pp. 140-147, (2002). (Data from: www.cs.sfu.ca/~colour)

[10] <http://www.mathworks.com/>

[11] F. Ciurea and B. Funt, “A Large Image Database for Color Constancy Research,” *Proc. IS&T/SID Eleventh Color Imaging Conference, Society for Imaging Science and Technology*, pp. 160-163, (2003)

[12] S.D. Hordley, and G.D. Finlayson, “Re-evaluating colour constancy algorithms,” *Proc. 17th ICPR*, Vol. 1, pp. 76-79, (2004)

[13] R.L. Eubank, “Spline Smoothing and Nonparametric Regression,” Marcel Dekker, New York, (1988)

| Method | SVR Dimension/ Norm Power | Median Angle | RMS Angle | Max Angle | Median Dist ($\times 10^2$) | RMS Dist ($\times 10^2$) | Max Dist ($\times 10^2$) |
|---------|------------------------------|-----------------|--------------|--------------|----------------------------------|-------------------------------|-------------------------------|
| GSI | | 3.91 | 10.11 | 33.79 | 2.71 | 7.15 | 22.65 |
| SVR | 2D | 4.65 | 10.06 | 22.99 | 3.41 | 7.5 | 16.41 |
| | 3D | 2.17 | 8.069 | 24.66 | 3.07 | 6.3 | 16.03 |
| SoG | 6 | 3.97 | 9.027 | 28.70 | 2.83 | 6.21 | 19.77 |
| Max RGB | | 6.44 | 12.28 | 36.24 | 4.46 | 8.25 | 25.01 |
| GW | | 7.04 | 13.58 | 37.31 | 5.68 | 11.12 | 35.38 |

Table 1 Comparison of GSI performance to that of 2D and 3D Support Vector Regression, Shades of Grey, Max RGB, and Grayworld. The results involve real-data training and testing on the 321 SONY images. Errors are based on leave-one-out cross-validation evaluation and are reported in terms of both the RMS angular chromaticity and distance error measures.

| | GSI | 2D SVR | 3D SVR | SoG (norm power = 6) | Max RGB | GW |
|----------------------|-----|--------|--------|-------------------------|---------|----|
| GSI | | = | - | = | + | + |
| 2D SVR | = | | - | = | + | + |
| 3D SVR | + | + | | + | + | + |
| SoG (norm power = 6) | = | = | - | | + | + |
| Max RGB | - | - | - | - | | - |
| GW | - | - | - | - | + | |

Table 2 Comparison of the different algorithms based on Wilcoxon signed-rank test on 321 images. A ‘+’ means the algorithm listed in the corresponding the row is better than the one in corresponding column; a ‘-’ indicates the opposite; an ‘=’ indicates that the performance of the respective algorithms is statistically equivalent.

| Method | Angular Degrees | | | Distance ($\times 10^2$) | | |
|---------|-----------------|-------|-------|----------------------------|------|-------|
| | Median | RMS | Max | Median | RMS | Max |
| GSI | 5.46 | 7.95 | 38.71 | 4.15 | 6.23 | 31.93 |
| 3D SVR | 4.91 | 7.03 | 24.80 | 3.62 | 5.16 | 18.62 |
| SoG | 6.71 | 8.93 | 37.01 | 4.83 | 6.59 | 27.99 |
| MAX RGB | 9.65 | 12.13 | 27.42 | 6.86 | 8.80 | 21.72 |
| GW | 6.82 | 9.66 | 43.84 | 5.25 | 7.82 | 45.09 |

Table 3 Comparison of GSI error to 3D SVR, SoG, Max RGB, and Grayworld. The results involve real-data training and testing on disjoint sets of 7,661 images taken from the Ciurea data set.

| Method | GSI | 3D SVR | SoG (norm power = 6) | MAX | GW |
|----------------------|-----|--------|----------------------|-----|----|
| GSI | | - | + | + | + |
| 3D SVR | + | | + | + | + |
| SoG (norm power = 6) | - | - | | - | = |
| MAX | - | - | + | | + |
| GW | - | - | = | - | |

Table 4 Algorithm comparison using the Wilcoxon signed-rank test for real-data training and testing on disjoint sets of 7,661 images from the Ciurea data set. Labeling ‘+’, ‘-’, ‘=’ as for Table 2.