

Automatic White Balance for Image Pipeline of Digital Still Camera

Yu-Mao Lin (林裕賢), Kun-Yi Chen (陳坤毅), and Chiou-Shann Fuh (傅楸善)

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan

ABSTRACT

In recent years, there has been a considerable growth in the sale and use of digital cameras. Consumers expect the quality of image more. White balance is one of the factors to improve the quality of image. The basic white balance methods are powerless to handle all possible situations in the captured scene. Our method tries to find the color temperature of the captured image and assigns the gain value corresponding to the color temperature. We use raw data of SONY DSC (Digital Still Camera) F-828 to do the image pipeline and implement this white balance method on it.

1. INTRODUCTION

The color of a scene in an image appears different depending on the illumination under which the image is taken. This is due to the fact that the different illumination sources, such as daylight, incandescent light, fluorescent light, have different power spectral distributions. Hence, as a result, an object may appear to have a color different than its true color. The white balance algorithm wants to make the white object be always white under different illumination.

Color temperature is a simplified way to characterize the spectral properties of a light source. When a white object is illuminated with low color temperature light source, the object in the captured image will have a reddish color. Similarly, if the white object is illuminated with high color temperature, the object in the captured image will have a bluish color. For developing white balance, it is necessary to know the illumination of captured image.

Keyword: white balance, gray world assumption, perfect reflector assumption, image pipeline.

2. BACKGROUND

The traditional methods used to adjust white balance automatically are mentioned below,

2.1. Gray World Assumption (GWA)

This method takes an image and scales its red, green, and blue color components such that the average

intensities of red, green, and blue components are adjusted to the same value.

2.2. Perfect Reflector Assumption (PRA)

This method locates the white point by finding the pixel with greatest luminance value and performs the white balance adjustment according to the reference white point.

3. LIGHT CONDITION DETECTION

In this section we will show how our algorithm detects various light conditions.

By analyzing five image with a Macbeth color checker taken under different light sources (Day light, Cool-white, TL-84 light, A light, H light), the distribution plots (in G/R, G/B space) are shown in Figure 1.

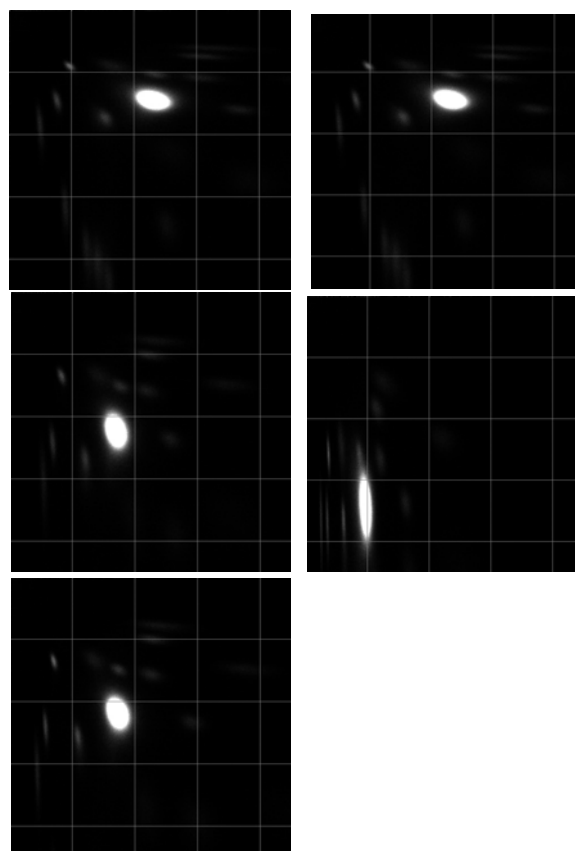


Fig. 1. Distribution plots of different light conditions

As the plots show, there is a dense area in every plot and we know that it represents the set of white patch and neutral patches of Macbeth color checker since they occupy most of the area on the Macbeth color checker. Consequently, if a given image's distribution plot looks like one of the plots of the five light conditions, then the image is highly possible to be taken under that light condition.

As a result, our algorithm uses similar idea to analyze a given image and detects its closest light condition.

3.1. Build-up Light Source Database

The first step of our algorithm is to analyze the characteristic of white point under different light sources. We take five pictures of Macbeth color checker under different scene types (Day light, Cool-white, TL-84 light, A light, and H light) in a light box, and under different sunlight conditions such as sunny, afternoon, cloudy, rainy, night, and indoor.

In each image, the left-bottom patch of the Macbeth color checker (which represents white color) is analyzed by counting the average of each channel (red, green, blue), to obtain the value: $R_{gain} = (G_{avg} / R_{avg})$, $B_{gain} = (G_{avg} / B_{avg})$. Then, the vector (R_{gain}, B_{gain}) is used as the representative of this light temperature.

For example, applying our algorithm to Sony DSC F-828's raw image, we get the representative vectors as following: Day light: (2.27, 1.46), Cool-white: (1.65, 2.25), TL-84 light: (0.93, 3.5), A light: (1.15, 2.85), H light: (1.67, 2.2), as shown in Fig. 2.

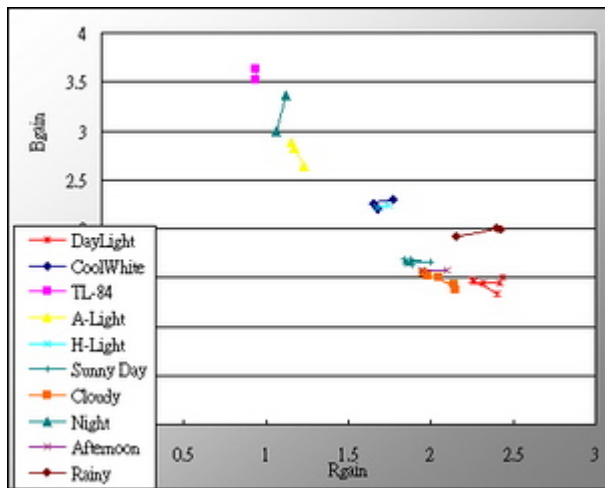


Fig. 2 Gain distribution of different light sources.

3.2. Light Condition Detection

After the database is built, we are able to analyze a given image and detect its light condition. The flowchart of light condition detection is shown in Fig. 3. For all pixels of a given image, our algorithm calculates Y , R_G , and B_G of this pixel by Equations 1.1 to 1.3.

$$Y = 0.299*red + 0.587*green + 0.114*blue \quad (1.1)$$

$$R_G = green / red \quad (1.2)$$

$$B_G = green / blue \quad (1.3)$$

Then, this pixel is detected to be a white point under a light condition if it's Y , R_G , and B_G satisfy Equation 1.4 to 1.6

$$5\% \text{ of } max_gray_value < Y \quad \& \quad Y < 60\% \text{ of } max_gray_value \quad (1.4)$$

$$R_{gain} - 0.05 < R_G \leq R_{gain} + 0.05 \quad (1.5)$$

$$B_{gain} - 0.05 < B_G \leq B_{gain} + 0.05 \quad (1.6)$$

If this pixel is detected to be a white point under a light condition, the counter of the light condition adds one. After all the pixels of the image have been calculated, the light source with the maximum counter value will be chosen, and it will be the detected light condition of the given image.

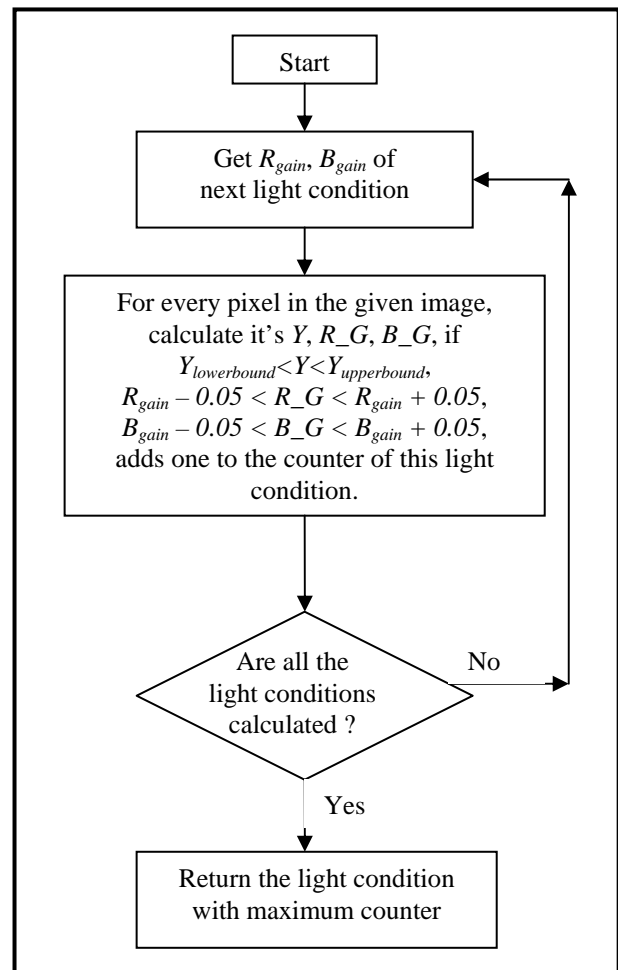


Fig. 3 Flowchart of light condition detection

4. ASSIGN CHANNEL GAINS

For a detected light condition, if we directly apply the channel gains as: $(R_{gain}, 1, B_{gain})$ to a image with a color checker, the white patch of the image will be purely white. For human's visual system, however, the eyes still sense the color cast of different light sources on a white object. For instance, a white board under a tungsten light appears to be much more yellow than

under a fluorescent light. Therefore, we believe that it is better to apply customized, pre-calculated channel gains to a known light condition.

In our algorithm, the gains are used as shown in Table. 1.

	Red gain	Green gain	Blue gain
Day light	1.8	1.0	1.62
Cool-white	1.72	1.0	2.3
TL-84 light	1.47	1.0	2.1
A Light	1.43	1.0	2.22
H Light	1.72	1.0	2.19
Sunny	1.87	1.0	1.67
Cloudy	1.92	1.0	1.62
Afternoon	1.85	1.0	1.6
Rainy	1.82	1.0	1.82
Night	1.3	1.0	2.3
Indoor	2.12	1.0	1.6

Table. 1. Gains look-up table

5. RESULTS

We use raw image data of SONY DSC F-828 to do the image pipeline and white balance. We choose GWA and PRA to compare with our white balance algorithm. The results are shown in Fig. 4 and Fig. 5.



GWA



PRA



Our method

Fig. 4. Result images.



GWA



PRA



Our method

Fig. 5. Result images.

6. REFERENCES

- [1] Y. C. Cheng, W. H. Chan, and Y.Q. Chen, "Automatic White Balance for Digital Still Camera," *IEEE Transactions on Consumer Electronics*, Volume 41, pp. 460-466, 1995.
- [2] J. Chiang and F. Al-Turkait, "Color Balancing Experiments with the HP-Photo Smart-C30 Digital Camera," PSYCH221/EE362 course project, Department of Psychology, Stanford University, 1999.
- [3] Varsha Chikane, "Automatic White Balance for Digital Still Camera," Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, 2004.
- [4] N. Kehtarnavaz, H-J oh, S. Reis, Y. Yoo, R. Talluri, "A Scoring Approach to Automatic White Balancing,".
- [5] M. Fedor, "Approaches to Color Balancing," PSYCH221/EE362course project, Department of Psychology, Stanford University, 1998.