# Automatic White Balance with Color Temperature Estimation

Po-Min Wang and Chiou-Shann Fuh
Digital Camera and Computer Vision Laboratory,
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, R.O.C.

*Abstract*--**This paper discusses automatic white balance in digital camera. Auto white balance is the most important image processing step in the image pipeline of digital camera, since it controls the color appearance of the image.**

## I. INTRODUCTION

Image pipeline is a technique to transform sensor's raw data to final compressed image, and automatic white balance (AWB) is the major step in image pipeline to correct the image's color. Without auto white balance adjustment, the photo will become bluish under high color temperature and have reddish color cast under low color temperature. In this paper, we propose a novel and robust auto white balance method based on color temperature estimation.

## II. THE ALGORITHMS

In our method, we use color temperature estimation by applying gray point voting approach. There are three steps in our method: (a) preliminary preparation, (b) gray point sieving and color temperature scoring, and (c) color temperature estimation and white balance adjustment.

### *A. Preliminary Preparation*

First, we capture GretagMacbeth ColorChecker under different standard light sources in light box, where the color temperatures of light sources should be from about 2500K to about 7000K, and increment of about 1000K. In our experiment, we take these photos under color temperatures 2840K, 3750K, 4082K, 4780K, 6000K, and 7070K, and we dot the neutral blocks (from white to black) of GretagMacbeth ColorChecker distribution of these captured raw data on *Cb*, *Cr* coordinates as shown in Fig. 1.

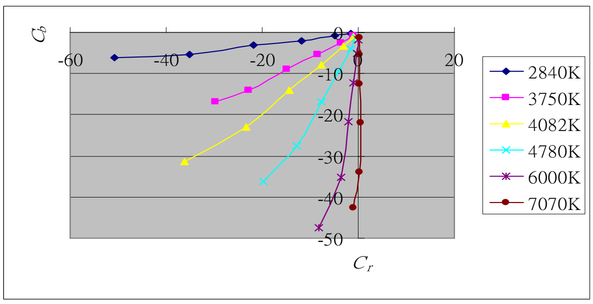


Fig. 1 Dots distribution of neutral blocks in GretagMacbeth ColorChecker under different color temperatures.

There are some characteristics in above Fig. 1, (a) Dots distribution of neutral patches in *Cb*, *Cr* coordinates form a line under any color temperature. (b) From left to right, the color temperature is from low to high. (c) Every line nearly passes through the origin. (d) Every color temperature has the same characteristic that the block with more brightness will be farther from the origin, i.e. the white patch is farther from the origin, and the black patch is nearer the origin.

We can see the relationship with illumination value (*Y*) and the summation of absolute value of *Cr* and *Cb* in Fig. 2.

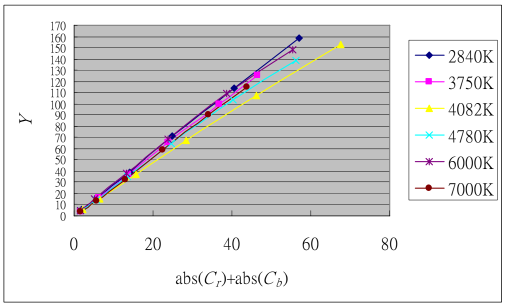


Fig. 2 The relationship with illumination value and the summation of absolute values of *Cr* and *Cb*.

Since these neutral patches in the same color temperature lie on a line as shown in Fig. 2, we use "best-fitting" line approach [4] to fit these data, and we could get these best-fitting line formulas for every color temperature.

After getting these best-fitting line equations, another preparation is to define the domain for these defined standard color temperatures on *Cb*, *Cr* space. First, we calculate the average *Cb*/*Cr* of blocks 19 to 23 in GretagMacbeth ColorChecker for every color temperature. Second, according to the average *Cb*/*Cr*, we divide the zone between every adjacent color temperature into three equal parts. We assign the upper (left) part of the three parts for lower color temperature, and the lower (right) part of the three parts for higher color temperature. Note that for the boundary color temperature (2840K and 7070K), we should copy the given domain and add it to the other side to be the boundary color temperature's domain.

Finally, we can generate each predefined color temperature's domain as shown in Fig. 3.

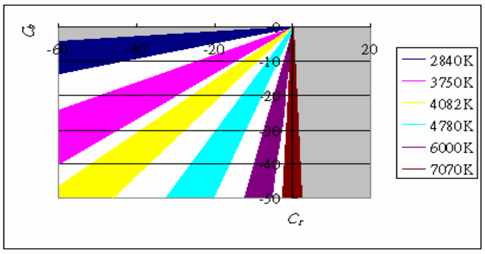


Fig. 3 Every predefined color temperature's domain.

### B. Gray Point Sieving and Color Temperature Scoring

We judge whether the pixel enters the predefined standard color temperatures' domain as shown in Fig. 3. If the point enters these predefined domains, we apply its summation of absolute values of *Cr* and *Cb* to the domain owner's best-fitting line equation, then, we can get the standard *Y* value ($Y_{std}$). We compare the *Y* component of the point with the standard *Y* value, and it should satisfy (1) to be the voter.

$$Y_{std} - 5 < Y < Y_{std} + 5 \qquad (1)$$

In addition, if the point does not enter the predefined domain, but it enters the white zone in Fig. 3, it also might be taken as the voter. We choose the maximum slope and minimum slope of these pre-calculated best-fitting lines as the upper bound and lower bound and judge whether it is within the upper bound and lower bound or not. If the answer is "yes", we take the pixel as gray point, else it would be ignored

By above gray point sieving, we can detect all gray points from the given image, and we can use these gray points to start the color temperature voting process. First, we calculate the *Cb*/*Cr* value of the pixel, if its *Cb*/*Cr* value enters one of these predefined standard color temperature predefined domain, the corresponding color temperature will be voted by one, else if the Cb/Cr value does not enter one of these defined color temperature domain, the two neighbor standard color temperatures will both voted by 0.5. After scoring process, every defined color temperature has its own scoring.

### C. Color Temperature Estimation and White Balance Adjustment

We detect the color temperature which has the maximum scoring and get its neighborhood color temperature. Now we should compare the highest-vote color temperature's votes with its neighborhood's votes. If the highest-vote color temperature is the boundary defined color temperature, i.e. the 2840K and 7070K, we detect whether the score of its neighborhood color temperature (3750K and 6000K respectively) satisfies (2).

$$Maxvote\_Neighbor * 2.5 \geq Max\_vote \qquad (2)$$

where *Max_vote* is the votes of highest-vote color temperature; and *Maxvote_Neighbor* is its neighborhood's votes.

If the score of the two color temperatures conform to (2), we can calculate the estimated color temperature and the scale factors for the given image by linear interpolation based on their score, otherwise, we choose the highest-vote color temperature as our estimated color temperature, and use the predefined corresponding scale factors.

If the highest-votes color temperature is not the boundary defined color temperature, i.e. 3750K, 4082K, 4780K, and 6000K, they have two neighbors. We judge whether the votes of the two neighbors are close enough by (3).

$$HighVoteNeighbor \leq LowVoteNeighbor * 1.3 \qquad (3)$$

where *HighVoteNeighbor* and *LowVoteNeighbor* are neighbors of hightest-vote color temperature, and *HighVoteNeighbor* has more votes than *LowVoteNeighbor*.

If the votes of the two neighbors satisfy (3), we pick the highest-vote color temperature as the estimated color temperature, and take the corresponding scale factors as the applied scale factors.

Otherwise, if the votes of the two neighbors do not satisfy (3), we can calculate the estimated color temperature and the scale factors for the given image by linear interpolation based on their score. After getting scale factors, we apply these scale factors to the given image to get a white balanced image.

## III. EXPERIMENTAL RESULTS

We tested our method against Chikane's method [3], gray world assumption method (GWA) [2], specular reflection assumption (SRA) [1], and Lin and Chen's white balance method (LCWB) [5] under different scenes. We calculate average of chromaticity values, $C = \sqrt{C_b^2 + C_r^2}$ for the achromatic patches of the GretagMacbeth ColorChecker. The results of objective evaluation are summarized in Table I.

| **Image** | Objectively Evaluated Value | | | | |
|---|---|---|---|---|---|
| | **GWA** | **SRA** | **CWB** | **LCWB** | **Our Method** |
| 01 | 22.5309 | 31.6000 | 22.4458 | 9.81679 | **8.46508** |
| 02 | 11.7917 | 40.4073 | 10.9576 | 26.8602 | **10.94436** |
| 03 | 17.6216 | 49.3232 | 17.4602 | 9.38136 | **8.05618** |
| 04 | 32.1942 | 49.6376 | 31.9432 | 4.38297 | **2.57258** |
| 05 | 16.8874 | 48.3714 | 16.9558 | 14.1370 | **12.41916** |
| 06 | 27.1639 | 49.7436 | 27.1639 | 10.8412 | **9.54723** |
| 07 | 28.7954 | 56.6631 | 28.8819 | 19.5732 | **16.17166** |
| 08 | 31.5152 | 18.9057 | 31.5699 | 21.3680 | **8.32726** |

Table I Chromaticity Value.

## REFFERENCES

[1] F. Al-Turkait, "Specular Reflection," http://ise.stanford.edu/class/psych221/projects/99/fahad/Spec_Ref_Intro.html, 1999.

[2] J. Chiang, "Gray World Assumption," http://ise.stanford.edu/class/psych221/projects/99/jchiang/intro2.html, 1999.

[3] V. Chikane and C. S. Fuh, "Automatic White Balance for Digital Still Camera," Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, 2004.

[4] S. H. Friedberg, L. E. Spence, and A. J. Insel, Linear Algebra, 2rd Ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.

[5] Y. M. Lin, K. Y. Chen, and C. S. Fuh, "Automatic White Balance for Image Pipeline of Digital Still Camera," Proceedings of IPPR Conference on Computer Vision, Graphics, and Image Processing, Wanli, Taiwan, A6-5, pp. 1544-1546, 2005.