Auto White Balance System Using Adaptive Color Samples for Mobile Devices

Wonwoo Jang[1], Kyoungsoo Son[2], Joohyun Kim[2], and Bongsoon Kang[1*]

[1]Multimedia Research Center, Department of Electronic Engineering, Dong-A University
840 Hadan-2-Dong, Saha-Gu, Busan 604-714, South Korea
[2]SAMSUNG Electro-Mechanics Co. Ltd.
314 Maetan-3-Dong, Young-Tong-Gu, Suwon 443-743, South Korea
*e-mail : bongsoon@dau.ac.kr

Abstract— **This paper presents a novel auto white balance (AWB) algorithm for images based on an YCbCr color space. The proposed algorithm divides all the pixels into fourteen groups by Y luminance and CbCr color components. For accuracy, we employ an adaptive color samples that can obtain the sum of the accumulated pixels up to the standard group for estimating a degree of white object's color changes. The color samples are used for the adjustment of the relative amounts of red, green, and blue primary colors. In order to avoid a field-flicker noise of video streams, the proposed algorithm adopts infinite impulse response (IIR) filters that vary the operating speed of overall AWB system. The performance of the algorithm is also verified by implementing the hardware of the system.**

I. Introduction

The sun rays, fluorescent and incandescent bulbs light the white objects; thereby people can see white objects. Although captured white objects appear natural under sun rays, they look bluish under fluorescent bulbs, and reddish under incandescent bulbs. The color temperature of a light source can be determined by comparing its chromaticity with a theoretical black-body radiator [1]. In spite of different color temperatures of the source under different environments, people perceive the colors of the source to be same by their memories. But, in capturing devices, the spectral distributions of the light sources are different, so captured white objects appear different.

We should know a white criterion that measures the difference between the represented image and the captured object. There have been many studies for the criterion [1-4]. Among them, a White Patch algorithm finds the white estimation by using each RGB for input image, a Neural Networks algorithm obtains the white estimation by using two trained neural networks, and a Grayworld algorithm obtains the white estimation by determining the averaged gray from RGB values [5-7].

This paper proposes a method by multiplying adjusted coefficients of color gains to the captured data of the white objects. This processing is called the automatic white balance (AWB). The AWB can be used as one of the most important functions on every mobile application device such as mobile phone cameras, digital still cameras, and so on. Without the AWB, a white object will not look like white through the capturing devices. This paper also uses adaptive color samples for adjusting the white by fourteen different groups.

II. Proposed Alogrithm

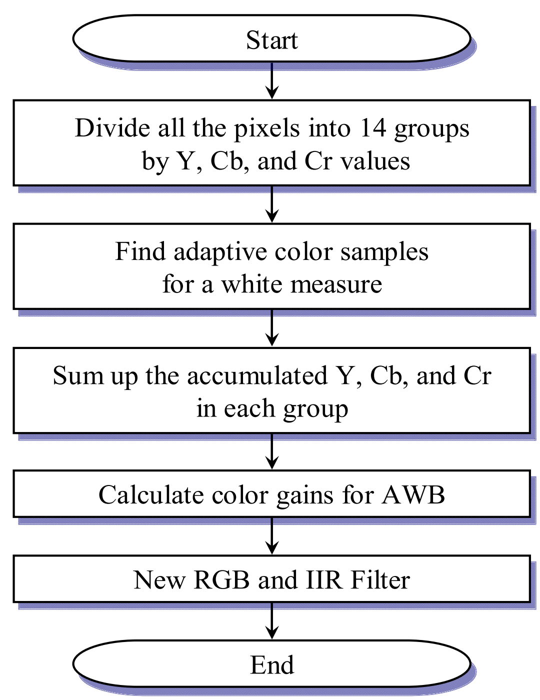


Figure 1. The flowchart of the proposed AWB algorithm

978-1-4244-2342-2/08/$25.00 ©2008 IEEE.

In the proposed AWB, we average the color difference signals (e.g., CbCr, UV or IQ), and then determine the color gains of R and B by using the averaged values. By doing so, the system uses 24bit YCbCr data to calculate the color gains. Figure 1 shows the flowchart of the proposed AWB algorithm.

We firstly divide all the pixels into fourteen groups by using three comparisons that are $Y_1 < Y_{in}$, $CbA_1 < Cb_{in} < CbB_1$, and $CrA_1 < Cr_{in} < CrB_1$. Eq. (1) shows the procedure:

$$\begin{aligned}
&if((Y_1 < Y_{in})\&(CbA_1 < Cb_{in} < CbB_1)\&(CrA_1 < Cr_{in} < CrB_1))\\
&\qquad TY_1 = TY_1 + Y_{in};\ \ TCb_1 = TCb_1 + Cb_{in};\\
&\qquad TCr_1 = TCr_1 + Cr_{in};\ \ N_1{+}{+};\\
&elseif((Y_2 < Y_{in})\&(CbA_2 < Cb_{in} < CbB_2)\&(CrA_2 < Cr_{in} < CrB_2))\\
&\qquad TY_2 = TY_2 + Y_{in};\ \ TCb_2 = TCb_2 + Cb_{in};\\
&\qquad TCr_2 = TCr_2 + Cr_{in};\ \ N_2{+}{+};\\
&\qquad\qquad\vdots\\
&elseif((Y_{14} < Y_{in})\&(CbA_{14} < Cb_{in} < CbB_{14})\&(CrA_{14} < Cr_{in} < CrB_{14}))\\
&\qquad TY_{14} = TY_{14} + Y_{in};\ \ TCb_{14} = TCb_{14} + Cb_{in};\\
&\qquad TCr_{14} = TCr_{14} + Cr_{in};\ \ N_{14}{+}{+};\\
&else\quad \text{"idle"}
\end{aligned} \tag{1}$$

where Y_{in}, Cb_{in}, and Cr_{in} denote the component values of YCbCr data in each pixel. Y_1, Y_2, …, and Y_{14} represent the Y luminance threshold values. CbA_1, CbB_1, CbA_2, CbB_2,…., CbA_{14}, and CbB_{14} represent two sets of Cb threshold values. CrA_1, CrB_1, CrA_2, CrB_2, …, CrA_{14}, and CrB_{14} represent two sets of the Cr threshold values. TY_1, TCb_1, and TCr_1 are the accumulated values of the YCbCr in the *G1* group. N_1 is the number of pixels in the group. TY_2, TCb_2, and TCr_2 are the accumulated values of the YCbCr in the *G2* group. N_2 is the number of pixels in the group. Likewise TY_{14}, TCb_{14}, and TCr_{14} are the accumulated values of the YCbCr in the *G14* group. N_{14} is the number of pixels in the group. The *"idle"* represents the pixels excluded from 14 groups.

We secondly find the adaptive color samples for a white measure in summated number SN_k. For this purpose, let's define

$$(N_1 + N_2 > N_3 + N_4 + N_5)\&(N_1 > N_2 + N_3) \tag{2}$$

$$SN_k > N_{color} \tag{3}$$

$$SN_k = \sum_{i=1}^{k} N_i \tag{4}$$

where N_{color} is about 20 percentage of input pixels to reduce a hardware complexity. In order to find adaptive color samples, we judge a particular image by the number of pixels in each group. To do so, we use Eq. (2) as the criterion. When Eq. (2) is met, we judge the input image to be normal. Otherwise the input is assumed to be abnormal. For the normal, we increase k value from 1 to 14 until SN_k of Eq. (4) is larger than N_{color} in Eq. (3). For the abnormal, we increase k value from 2 to 14 until SN_k is larger than N_{color}.

We thirdly sum up the accumulated values of TY_i, TCb_i, and TCr_i in each group with $i = 1 \sim k$. Eqs. (5)-(7) show the procedure:

$$STY_k = \sum_{i=1}^{k} TY_i \tag{5}$$

$$STCb_k = \sum_{i=1}^{k} TCb_i \tag{6}$$

$$STCr_k = \sum_{i=1}^{k} TCr_i \tag{7}$$

where STY_k , STY_k, and $STCr_k$ denote the summation from 1 to k.

We fourthly calculate the color gains to adjust the AWB. Let's define the average values of three component signals of YCbCr as

$$Y_{ave} = \frac{STY_k}{SN_k},\quad Cb_{ave} = \frac{STCb_k}{SN_k},\quad Cr_{ave} = \frac{STCr_k}{SN_k} \tag{8}$$

where Y_{ave}, Cb_{ave}, and Cr_{ave} denote the averages. We now use the conversion equations to derive RGB values that are the adaptive color samples [1]:

$$\begin{aligned}
R_s &= 1.164\times(Y_{ave}-16)+1.596\times(Cr_{ave}-128)\\
G_s &= 1.164\times(Y_{ave}-16)-0.813\times(Cr_{ave}-128)-0.391\times(Cb_{ave}-128)\\
B_s &= 1.164\times(Y_{ave}-16)+2.018\times(Cb_{ave}-128)
\end{aligned} \tag{9}$$

We can now calculate the color gains for AWB. They are

$$R_{gain} = \frac{G_s}{R_s},\quad G_{gain} = \frac{G_s}{G_s} = 1,\quad B_{gain} = \frac{G_s}{B_s} \tag{10}$$

where R_{gain}, G_{gain}, and B_{gain} represent R, G, and B color gains for AWB.

We can finally adjust all the RGB values by multiplying the color gains to the RGB inputs:

$$R_{new} = R_{gain}\times R_{in},\quad G_{new} = G_{in},\quad B_{new} = B_{gain}\times B_{in} \tag{11}$$

where R_{in}, G_{in}, and B_{in} denote the input values of RGB. R_{new}, G_{new}, and B_{new} represent the adjusted RGB data after the AWB proposed in this paper. In order to avoid frame flickers caused by abrupt image changes in consecutive frames [8], we use an IIR filter with variable time constants. We use

$$H(z) = \frac{b_0}{1 - a_0 Z^{-1}} \tag{12}$$

where a_0 and b_0 indicate the filter coefficients. Figure 2 shows the time responses of the filter. The fastest and slowest of the time response are denoted as ⓐ and ⓑ. ⓐ shows that three frames are required to arrive 90 percentage of the maximum, and ⓑ requires 45 frames likewise.

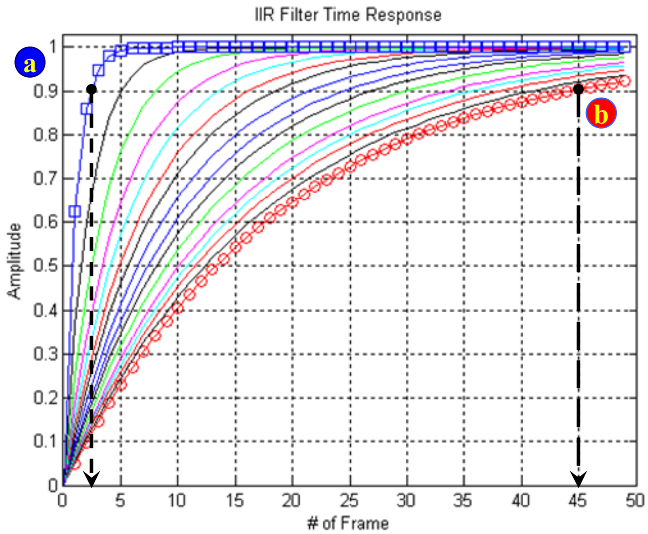


Figure 2. The time response of IIR filter

III. HARDWARE ARICHTECTURE

Figure 3 shows the simplified block diagram of an image signal processor that can be used in mobile applications. It is comprised of five building blocks. To verify the performance of the proposed algorithm, the system is designed by using Verilog HDL. The Pre_Processing derives RGB values from the Bayer signals captured from the CMOS Sensor Lens. The Pre_AWB calculate final RGB values in Eq. (11). The AWB conducts the white balance in Eqs. (1)-(10). The Color Processing conducts various color processing such as color interpolation, gamma processing, and so on. The Post_Processing enhances the image quality and sends 16bit 422 YCbCr for the following display devices.

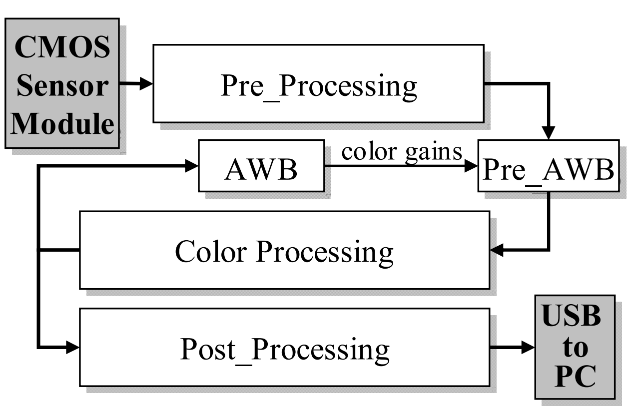


Figure 3. Simplified block diagram of an image signal processor

IV. SIMULATION AND EXPERIMENTAL RESULTS

Figure 4 shows are the simulation results of a standard color chart achieved by the proposed AWB algorithm. In order to measure the performance of the proposed system, we use Euclidean distances (ΔE^*_{ab}) in CIEL*a*b between the reference white and the white-balanced colors [9]. For an ideal white, the distance should be zero. The Euclidean distances without AWB in Figs. 5(a), (c), and (e) corresponding to the symbols ①, ②, ③, and ④ are larger than those with AWB in Fig. 5(b), (d), and (f), respectively. This reveals that the proposed AWB performs very well to achieve the white in images.

	①	②	③	④
ΔE^*_{ab}	8.664	11.526	10.707	8.927

(a)

	①	②	③	④
ΔE^*_{ab}	1.425	6.563	8.452	7.511

(b)

	①	②	③	④
ΔE^*_{ab}	19.928	18.111	14.743	9.867

(c)

	①	②	③	④
ΔE^*_{ab}	3.149	3.370	3.207	7.895

(d)

	①	②	③	④
ΔE^*_{ab}	6.185	14.136	14.191	11.725

(e)

	①	②	③	④
ΔE^*_{ab}	1.335	1.592	2.490	1.192

(f)

Figure 4. Simulation Results:
(a) D65 without AWB, (b) D65 with proposed AWB,
(c) Horizon without AWB, (d) Horizon with proposed AWB,
(e) Coolwhite without AWB, (f) Coolwhite with proposed AWB

Figure 5 shows two PCB verification boards for the proposed system. There are three major building devices of a CMOS sensor, a FPGA device, and a USB-to-PC interface.

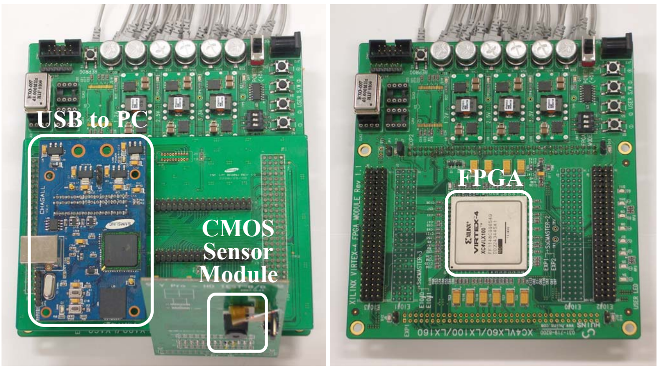
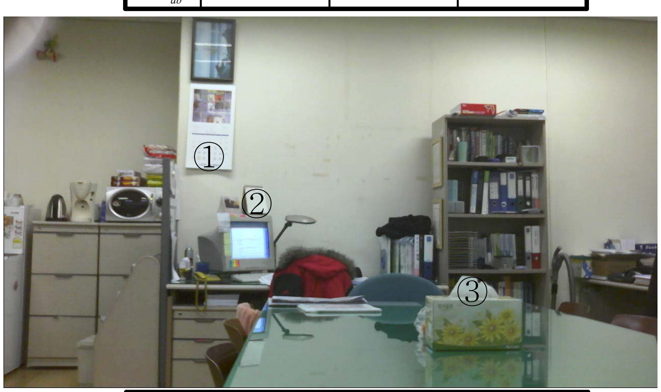


Figure 5. PCB demonstration boards

Figure 6 shows a compound image captured from the verification board. The Euclidean distances for ①, ②, and ③ in Fig. 6(a) are much larger than those in Fig. 6(b), respectively. Once again, we see that the proposed AWB performs very well to achieve the white in images.

(a)

	①	②	③
ΔE^*_{ab}	25.876	28.902	27.008



(b)

	①	②	③
ΔE^*_{ab}	2.105	2.857	3.439

Figure 6. Experimental Results: (a) Compound image without AWB and (b) Complex image with proposed AWB

V. Conclusions

In this paper, we proposed an AWB system based on YCbCr. We divided all pixels into fourteen groups to decide normal and abnormal images. We summed up the accumulated Y, Cb, Cr data, and the number of effective pixels to find adaptive colors samples. In order to avoid the frame flickers caused by abrupt image changes in consecutive frames, we additionally used an IIR filter with various time responses for following display devices. A means of the Euclidean distances to show the performance of the proposed system was adopted for verification. The distances with AWB were clearly much smaller than those without AWB. We thus conclude that the proposed AWB system can be applied to many mobile devices such as mobile phone cameras, digital still cameras, and so on.

Acknowledgment

This work has been supported in part by the 2nd BK21 funded by the Korean Government and Samsung Electro-Mechanics Co. Ltd. The authors also wish to thank the IDEC for its software assistance.

References

[1] R.C. Gonzalez, *Digital Image Processing*, 3rd ed., Addison Wesley Longman, 2007.

[2] B. Kang, O. Moon, C. Hong, H. Lee, B. Cho, and Y. Kim, "Design of Advanced Color-Temperature Control System for HDTV Applications," JKPS, Vol. 41, No. 6, pp. 865~871, Dec. 2002.

[3] Po-Min Wang, Chiou-Shanna Fuh, "Automatic White Balance with Color Temperature Estimation," 2007 Int. Conf. on Consumer Electronics, Jan. 2007.

[4] P. Green and L. MacDonald, *Colour Engineering-Achieving Device Independent Colour*, John Wiley & Sons Ltd., 2002.

[5] G. Buchsbaum, "A spatial processor model for object colour perception," Journal of Franklin Institute, vol.310, no. 1, pp. 1-26, 1980.

[6] V. Cardei, B. Funt, and K. Barnard, "Adaptive illuminant estimation using neural network," Proc. Of the 8th Int. Conf. on Artificial Neural Network, pp.749-754, Skvde, Sweden, 1998.

[7] E.H. Land, "The retinex theory of color vision,"Scientific American, pp. 108-129, 1997.

[8] H. Lee, H. Choi, B. Lee, S. Park, and B. Kang, "One Dimensional Conversion of Color Temperature in Perceived Illumination," IEEE Trans. On Consumer Electronics, Vol. 47, No. 3, pp. 340~346, Aug. 2001.

[9] R. Pete, *Digital Color Correction*, Thomson Learning, 2005.