Color Casts Detection and Adjustment

Sheng-Fuu Lin [1], Huang-Tsun Chen [2] and Tsung-Han Lin [3]

[1] Department of Electrical and Control Engineering, National Chiao Tung University
Hsinchu, Taiwan, R.O.C

[2] [1] Department of Electrical and Control Engineering, National Chiao Tung University
Hsinchu, Taiwan, R.O.C

[3] [1] Department of Electrical and Control Engineering, National Chiao Tung University
Hsinchu, Taiwan, R.O.C

Abstract
This paper presents a n ew method for detection and adjustment color cast. Using the neural network to detect color cast and classify images into three subsets: no cast, real cast, and intrinsic cast (image presenting a cast due to a predominant color that must be preserved). We have a database of 700 images which are downloaded from internet or acquired using various digital still cameras. We randomly select 350 images from the database for the neural network learning, and the others are for testing. From each training image, we can calculate 13 statistical parameters as input to the neural network. The second part is the white balance algorithm which is applied to the image while a real cast is found by the color cast detector. The test image is divided into *m* blocks. For each block, the output weighting can be obtained by a fuzzy system and the luminance weighted value is also calculated. Finally, we can obtain the new amplifier gains of the *R*, *G*, and *B* channel to adjust the color cast. If the input image be classified as no cast or intrinsic cast, white balance algorithm is not applied.
Keywords: ***Cast detection; White balance; Neural network***.

1. Introduction

It is well known that the digital signal processing technology has been widely adopted in the popularity of the internet and the rapid growth of computer industry; everyone comes across digital images anytime in daily life. The image quality may not be best presented, not only due to the skill of the photographers and the capturing device, but also the light sources or background while shooting. As digital cameras are concerned, owing to the difference between human eyes and the lens, it is necessary to develop the techniques of automatic focus, automatic exposure, automatic color adjustment and white balance in order to improve the color qualities of the images.

Colors are essential features in digital images. A few common algorithms for color adjustment were proposed. The most widely used one is *white balance algorithm* which is applied to adjust the color cast caused by the light sources when capturing images. There are two categories, *gray world assumption* [1] and *max white method* [2]. Gray world assumption calculates the weighting values of color correction by matching the color average value and predefined reference gray value. However, it m ay fail to work if the image only contains a few color elements. Max white method derives the weighting values of color correction by adjusting the white point in the image to predefined reference white point. It fails if the white point is not found in the image.

Besides, there are *neural network method*s [3] which determine the type of illuminant and correct the images after collecting a vast amount of illuminant data. *Color by correlation method* [4] first collects illuminant data in order to set up *correlation matrix*, by which to determine the type of illuminant and correct the image. *Illuminant voting method* [5] applied probability and voting process to determine the type of illuminant. Neural network, color by correlation and illuminant voting methods are all complicated since they require more computation in advance.

Color cast detector functions as a protection for the images without color cast before white balance algorithm is applied. If the image is with color cast, color adjustment will be processed using white balance algorithm. Other methods are *threshold method* [6] and *histogram method* [7-9].

The system can be applied to the following cases：(1) images uploaded to or downloaded from the internet with color cast; (2) images with color cast caused by improperly adjusting automatic white balance in the digital cameras; (3) images with color cast from other devices, such as scanners or monitors; (4) old and stained photos recaptured by digital camera; (5) images with uncertain color cast.

Figure 1 demonstrates the flow chart of color cast detection and adjustment. There are five steps in the system. After reading the input image, a 13-dimensional feature vector is computed and serves as the input to train the neural network. The neural network determines if there is a color cast is formed. The input images will be made by classifying the type of color cast and color cast center, and then using fuzzy system with luminance weighting value to adjust the image color cast. If the input image does not contain a color cast, white balance algorithm is not necessarily applied.

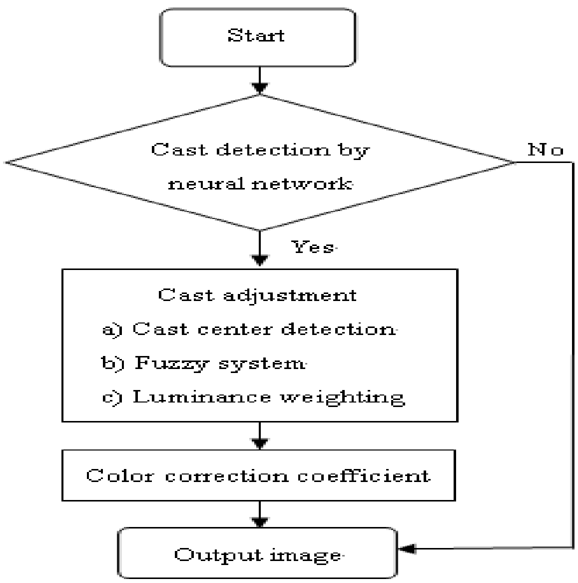


Fig. 1 System flow chart.

2. Color Cast Detection

The types of color cast in the images can be classified as no cast, real cast, or intrinsic cast. Because a great deal of area in the image is occupied by the same or similar colors, there exists the intrinsic cast in the image. If the test image is classified as having real cast, the white balance algorithm must be applied to the image. In general, white balance algorithms do not including the ability of cast detection. The difficulty of color cast detection is how to classify the intrinsic cast images from real cast images.

The nonlinear classification function of neural network can be used to classify the input images as real cast, no cast, and intrinsic cast. It is no need to proceed white balance for images having intrinsic cast or no cast, while white balance is necessary applied to images having real cast. Because the accuracy of the cast classification will affect the quality of output image, the main purpose is to distinguish the intrinsic cast from the real cast.

The input node of the neural network is respectively 13 features with representatives. Seven of the features are basic ones to detect real cast and no cast, while the other six are to detect intrinsic cast. The seven features are

$$Feature1 = \left[\bar{R} \ \ \bar{G} \ \ \bar{B} \ \ K \ \ D_{\sigma} \ \ Dis_{max} \ \ C\right] \tag{1}$$

where $\overline{R}$, $\overline{G}$, and $\overline{B}$ are the average values of red, green, and blue channels, respectively. The purpose is to determine the shift of a certain color channel. The value of K can be obtained from the threshold method and can determine general color cast. If $K > K_{threshold}$ this means color cast takes place. The greater K is, the stronger the cast is. D_{σ} mentioned in the histogram method, has similar functions as the above K value. $Dis_{\max}$ is the distance between the greatest peak and the axis, as shown in Fig. 2. C is the number of classifying peaks in a 2D histogram. Threshold τ is set first. If the peak value in the 2D histogram is greater than then threshold, the peak is projected on the *ab*-plane. If the peak value is smaller than the threshold, then the projection will not appear on the *ab*-plane. Figure 3(a) shows the 2D histogram. Figure 3(b) demonstrates the location on the *ab*-plane after the peaks are filtered τ. After localizing the peaks from Fig. 3(b), we classify the peaks so as to show the distribution of the peaks on the *ab*-plane. The results are shown in Fig. 4. If the number of groups of the peaks is great, this means the color cast is unlikely to occur. If the number of clusters of the peaks is small and $Dis_{\max}$ is great, it means real cast or intrinsic cast may take place. The method of classifying the location of the peaks is *Max-Min Distance* (MMD) [10]. The group number is the clustering number C of peak value needed in a 2D histogram.

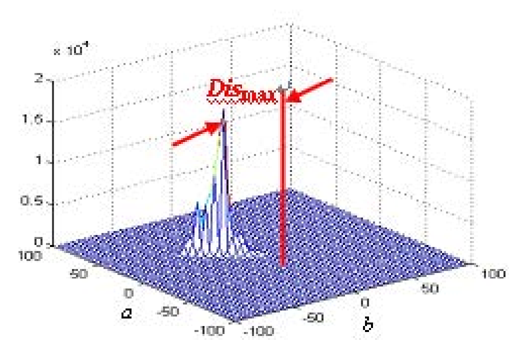
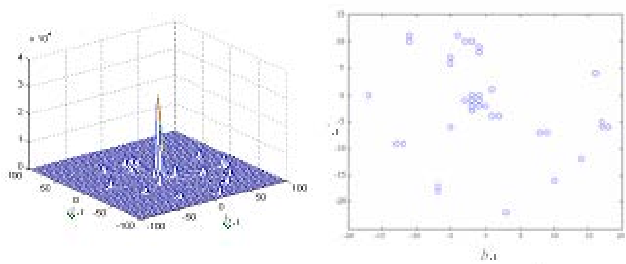


Fig. 2. $Dis_{\max}$ is the distance between the greatest peak and the axis.



(a) 2Dhistogram (b) the location on ab-plane

Fig. 3. 2D histogram.

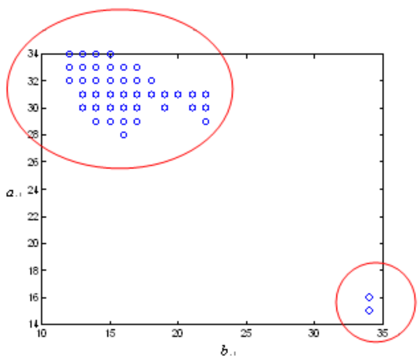


Fig. 4. classification of the peaks on *ab*-plane

Six feature vectors for detecting intrinsic cast are as follows:

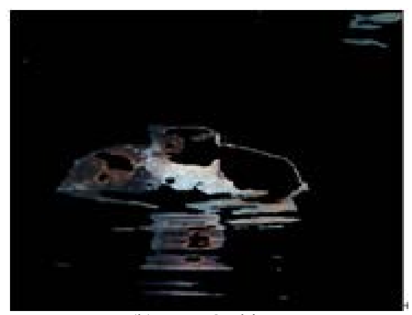
$$Feature2 = [R_{SD} \quad G_{SD} \quad B_{SD} \quad \tilde{D}_{\sigma} \quad \tilde{K} \quad \tilde{D}is_{max}] \tag{2}$$

where R_{SD}, G_{SD}, and B_{SD} are the standard deviation values of red, green, and blue color channels in the image, respectively. Standard deviation can indicate the similarity of the colors in the image. If the similarity of colors is high, it means strong color cast or intrinsic cast take place. $D_{\sigma NNO}$ is the value of D_{σ} in the NNO(near neutral objects) area proposed by [7] and it can determine color cast in the NNO area. The detection in NNO area may result from the removing of non-intrinsic color carelessly. A *mode color method* is proposed here. We first convert color of the input image into *YCbCr* chromacity coordinates by the equations

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb = -0.169R - 0.331G + 0.500B$$
$$Cr = 0.500R - 0.418G - 0.082B. \tag{3}$$

We can find out the peak of distribution in the input image, then make a circle with (*Cb, Cr*) of the peak as the center and set the radius *r*, where *r* is predefined value. The colors within the circle are the mode colors and we can remove these colors. The remaining colors will determine the color cast. Detecting NNO is faster but it may delete non-intrinsic color. The mode color method is more accurate but it takes more calculation time. Figure 5 demonstrates the comparison of the two methods. It is more accurate to use the mode color method to adjust the area of dominant color, therefore, $\tilde{D}_{\sigma}$ is used to substitute $D_{\sigma NNO}$ as the feature vector. $\tilde{K}$ is the parameter derived from the image processed by the mode color method and is used to determine the remaining colors with any color cast. $\tilde{D}is_{max}$ is the distance between the axis and the greatest peak value of the colors in the remaining area in the 2D histogram.

(a) Input image



(b) NNO object

(c) Mode color method

Fig. 5. Input image and it's NNO objects

Neural network applied here is back-propagation neural network which contains two hidden layers. There are thirteen neurons in the input layer of neural network, including seven basic features for detecting real or no cast images and six features for detecting intrinsic cast images. Thirty neurons are in the first hidden layer and fifteen neurons are in the second hidden layer. Three neurons in the output level each represent no cast, intrinsic cast, and real cast respectively. Each neuron contains sigmoid activation function.

3. Color Cast Adjustment

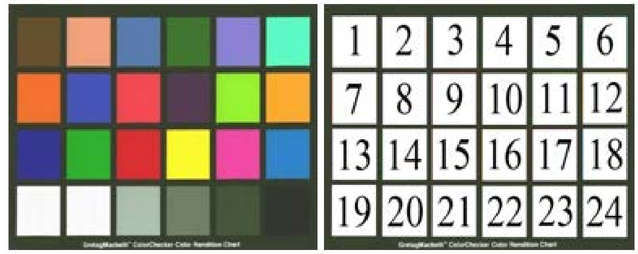
After classifying the type of color cast in the image, the image with real cast must be processed as a color cast adjustment step. We use white balance algorithm as a color cast adjustment tool to correct the undesired color shifts in the image. The colors of images recorded by a camera depend not only on the surface properties of the objects present in the scene depicted, but also on the light sources and characteristics of the image capture device. Unlike human vision, image-capturing devices (such as digital cameras) cannot adapt their spectral responses to cope with different lighting conditions; and thus as a result, the acquired image may have a cast. In general, when the light source color temperature is high, a blue cast will appear, and when the light source color temperature is low, a red cast will appear. With white balance algorithm, a digital still camera can automatically compensate for the color shifts due to the different color temperatures of the light sources by adjusting the RGB gains of the corresponding channel.

Take the Macbeth color chart in as Fig. 6 as the standard, and use the first nineteen common colors in the color chart to detect the color cast. First convert the chromaticity coordinates of the image from *RGB* to *CIEXYZ,* then convert the chromaticity coordinates from *CIEXYZ* to *CIExy*. The two converted functions are:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.7689 & 1.7517 & 1.1302 \\ 1.0000 & 4.5907 & 0.0601 \\ 0.0000 & 0.0565 & 5.5943 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{4}$$

and

$$x = \frac{X}{X+Y+Z}$$
$$y = \frac{Y}{X+Y+Z}. \tag{5}$$



(a) Color chart (b) Color patch numbers

Fig. 6. Macbeth color chart

Project the nineteen colors on the *CIE chromaticity diagram*, as shown in Fig. 7. Let $P_{x,y}(n)$ denote the projected coordinate values with respect to the nth color. Calculate the input image average value x_{avg}, y_{avg}, in *CIExy* color model and the distance $D(n)$ between (x_{avg}, y_{avg}) and $P_{x,y}(n)$. We can acquire the aberrant ratio $R(n)$ and cast center $W_{x,y}$ by $D(n)$:

$$R(n) = \frac{D(n)^{-1}}{\sum_{i=1}^{19}(D(i)^{-1})}, \quad n = 1,2,\ldots,19 \tag{6}$$

$$W_{x,y} = \sum_{n=1}^{19} R(n) P_{x,y}(n) \tag{7}$$

Convert the color cast center $W_{x,y}$ from the chromaticity coordinates *CIExy* to chromaticity coordinates *RGB* and then W_R, W_G, and W_B will be obtained. The color cast center will be one of the inputs of the fuzzy system.

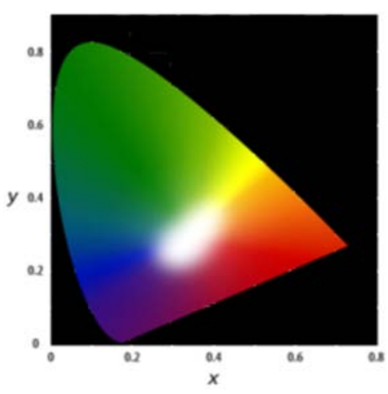


Fig. 7. CIE chromaticity diagram.

In the proposed white balance algorithm, the three color components *R*, *G*, and *B* are applied to the two inputs and one output fuzzy system, respectively. The first step is to calculate the two inputs of the system. Divide the image into *m* blocks, as shown in Fig. 8. Let *k* denote the *k*th block of the image. Respectively calculate standard deviation $SD_R(k)$, $SD_G(k)$, $SD_B(k)$ and the average values $\overline{R(k)}$, $\overline{G(k)}$, $\overline{B(k)}$ in RGB space for each block. As for red value, set input one of fuzzy system as $F_{input1}(k)$=$SD_R(k)$ and input two as $F_{input2}(k)=\left|W_R-\overline{R(k)}\right|$. W_R is the color cast center. When the standard deviation is small and the distance between average values of color components and cast center is also small, the color correction weighting must be large. On the other hand, when the standard deviation is large and the distance between average values of color components and cast center is also large, the color correction weighting must be small. By this, the weight of color correction $F_{R\_output}(k)$, $F_{G\_output}(k)$, and $F_{B\_output}(k)$ of the three colors will be obtained respectively.

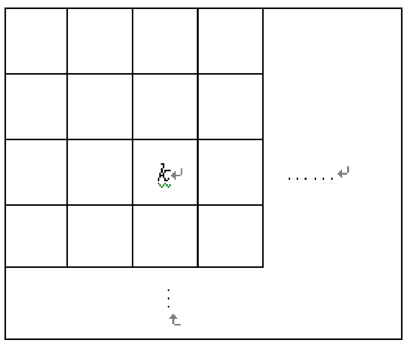


Fig. 8. The division of the image..

At high luminance, the colors are easily saturated; while at low luminance, the color components become colorless. The luminance weighting value will be set small if the average luminance of the block is under the conditions of high-end and low-end. We calculate the average luminance of block k, and obtain the luminance weighting value by using Eq. (8):

$$W_L(k)=\exp\left(\frac{-0.5(\overline{L(k)}-\mu)^2}{2\rho^2}\right) \tag{8}$$

where ρ is standard deviation. When $\overline{L(k)}=\mu$, $W_L(k)$ will be the maximum, weighting values calculated above are combined as Eqs. (9). Average weighting values of R, G, and B color channels will be obtained from:

$$FWA_R=\frac{\sum_{k=1}^{m}F_{R\_output}(k)\cdot W_L(k)\cdot\overline{R(k)}}{\sum_{i=1}^{m}F_{R\_output}(i)\cdot W_L(i)}$$
$$FWA_G=\frac{\sum_{k=1}^{m}F_{G\_output}(k)\cdot W_L(k)\cdot\overline{G(k)}}{\sum_{i=1}^{m}F_{G\_output}(i)\cdot W_L(i)}$$
$$FWA_B=\frac{\sum_{k=1}^{m}F_{B\_output}(k)\cdot W_L(k)\cdot\overline{B(k)}}{\sum_{i=1}^{m}F_{B\_output}(i)\cdot W_L(i)} \tag{9}$$

Then calculate the respective color correction coefficients of R, G, and B color channels by:

$$Weight_R=\frac{(FWA_R+FWA_G+FWA_B)/3}{FWA_R}$$
$$Weight_G=\frac{(FWA_R+FWA_G+FWA_B)/3}{FWA_G}$$
$$Weight_B=\frac{(FWA_R+FWA_G+FWA_B)/3}{FWA_B} \tag{10}$$

After adjustment, tri-stimulus values R_a, G_a, and B_a are:

$$R_a=Weight_R\cdot R$$
$$G_a=Weight_G\cdot G$$
$$B_a=Weight_B\cdot B \tag{11}$$

Finally, the color corrected output image will be done.

4. Experimental Results

Because a great deal of area in the image is occupied by the same or similar colors, the intrinsic cast exists in the image, as shown in Fig. 9. The neural network is a two-layer back-propagation neural network. The input layer consists of thirteen nodes (thirteen features obtained from the image), the first hidden layer has thirty nodes, the second hidden layer has fifteen nodes, and the output layer has three nodes. Each output node means there is one type of color cast (indicating no cast, real cast, or intrinsic cast). Initial weighting values and threshold values are random numbers between -1 to 1. Each neuron has its own sigmoid activation function as below:

$$sgm(A)=\frac{1}{1+\exp(-A)} \tag{12}$$

where A means the difference between the weighted sum and the threshold value of the input layer. To avoid increasing the convergence time of the neural network, $Dis_{\max}$ and $\widetilde{D}is_{\max}$ should be calculated using logarithm function. Training targets are designated by people. If real cast exists in the image, the target is set as (1,0,0). If intrinsic cast exists in the image, the target is set as (0,0,1). Finally, if there is no exist cast in the image, the target is set as (0,1,0).

Fig. 9. The images with intrinsic cast..

After training the neural network, we will obtain the output weighting values and threshold values. These values will be calculated with the features obtained from the input testing images to classify the type of cast in the input testing images. We have 350 training images and 350 testing images. The size of these images are not the same. These images are downloaded from the internet, captured by different digital still cameras, or scanners. The performance of detecting real cast and no cast images are similar using three methods, but the detection of intrinsic cast images using the proposed color cast detection method is better than the other two. This is because the nonlinear classification of the neural network is better than dichotomy of the threshold method and the histogram method which the predefined threshold value will affect the classification result is critical. Besides, the NNO region proposed by the histogram method is not reliable and mode color method proposed by this paper can more accurately preserve the non-dominant color.

Table 1
The comparison between three cast detection methods.

	Threshold method with $K = 1.1$	Histogram method with $D_{\sigma NNO}$ =0.6	Neural Network
Images with real cast (Macbeth color chart)	92.6%	91.3%	93.3%
Images with real cast (Nature image)	95.0%	93.3%	95.0%
Images with no cast (Nature image)	90.0%	87.0%	92.0%
Images with intrinsic cast (Nature image)	------	82.7%	91.1%

Although the proposed method has better performance than others, the unsuccessful cases showed two possible errors:

1) The method classifies a cast as a dominant color; the image is not further processed through the white balance algorithm. This is because strong cast exists in the image, or the dominant color has cast, as shown in Fig. 10.

Fig. 10. Images having real cast erroneously classified as having intrinsic cast..

2) The cast detector may consider a dominant color as a cast; as shown in Fig. 11, as the remaining area after deleting the mode color in the image has less information or less number of colors. Applying the white balance algorithm to these images may produce color changes in the output image.

Fig. 11. The system classifies a dominant color as a cast.

We use white balance algorithms to correct Macbeth color charts, including six different color casts, and compare the output images of white balance algorithms using the Euclidean distance ΔE^*_{ab} in the $CIEL*a*b*$ model :

$$\Delta E^*_{ab} = [(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2]^{1/2} \tag{13}$$

where

$$\begin{aligned} \Delta L^* &= L^*_{output} - L^*_{input} \\ \Delta a^* &= a^*_{output} - a^*_{input} \\ \Delta b^* &= b^*_{output} - b^*_{input}. \end{aligned} \tag{14}$$

and subscripts in Eqs. (14) *input* and *output* are used to index the original and corrected images, respectively. Note that L^*, a^*, and b^* can be obtained from

$$L^* = \begin{cases} 116 \times \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3 \times \left(\frac{Y}{Y_n}\right) & \text{if } \frac{Y}{Y_n} \le 0.008856 \end{cases}$$

$$a^* = 500 \times \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right)$$

$$b^* = 200 \times \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Z}{Z_n}\right) \right) \tag{15}$$

where

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > 0.008856 \\ 7.787 \times t + \frac{16}{116} & \text{if } t \le 0.008856 \end{cases}$$

X, Y, and Z are derived from Eq. (4), X_n, Y_n, and Z_n are reference white tri-stimulus values. To observe the relationship between the number of color patches and the white balance algorithms, we randomly pick up four patches from the standard Macbeth color chart to form the test color charts and calculate the average error. The test color charts including 8, 12, and 16 patches are formed the same way. To observe the relationship between the existence of white patch and white balance algorithms, the white patch will be removed from the test color charts. Nature images are photos captured in our daily lives and contain an abundance of colors such as skin, sky, or vegetation colors. The test images are images of nature captured under different light sources, such as daylight, neon, or tungsten bulb sources.

For Fig. 12, algorithms give fewer errors in large numbers of color patches on the chart than those in small numbers of color patches on the chart. For Fig. 13, the errors in all algorithms become higher when the white color patch is not included in the testing color chart. As for *advanced max white method* (AMW), the white color patch is the crucial reference in the algorithm. The algorithm does the correction according to the white color cast. However, the algorithm only aims at the white color for correction, which causes insufficient correction in other colors. When the white color patch is not included in the testing color chart, the algorithm still has to look for the reference white color. If the found reference white color is improper, it may cause correction in a failure. As for the *gray world assumption* (GWA), reference gray is the basis of color correction. The number of colors in the testing color charts is too limited; the reference gray calculated by GWA will not be "gray." That is to say, when chromaticity coordinates is converted to *CIEL*a*b** coordinates, the reference gray will not fall on the axis or the small area near the neutral axis. Weighting values of color correction obtained by the algorithm fails to recover the colors of the image in better quality, and sometimes in worse quality. As for *standard deviation weighted gray world* (SDWGW), the algorithm will divide the image into several blocks, calculating color standard deviation, average of each block and weighting average of standard deviation in order to obtain reference gray. The way that SDWGW modifies GWA is by using standard deviation to improve the defect caused by GWA when insufficient colors are in the image. However, when color standard deviation in each block of the image is small, the performance of this algorithm will degenerate to GWA. The happens when too many blocks in the image containing zero standard deviation values leads to failure from correction.

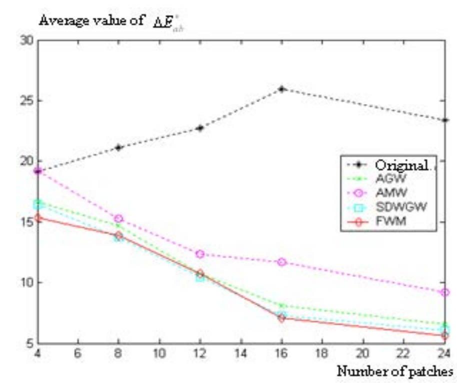


Fig. 12. The relation between the number of color patches and the average value of ΔE^*_{ab}

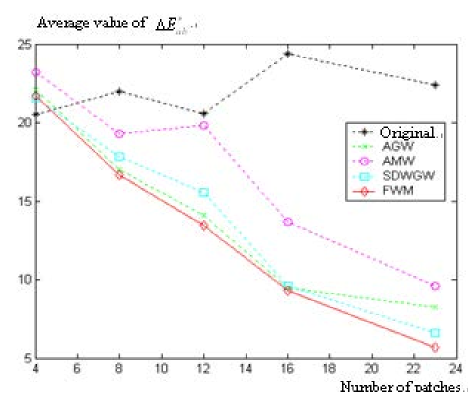


Fig. 13. The relation between the existence of white patch and the average value of ΔE^*_{ab}

The process of fuzzy system white balance algorithm is to divide the image into several blocks, to obtain color standard deviation and average value of each square and to get two inputs of fuzzy system based on color standard deviation, average and the distance of center of color cast. The purpose of applying the two inputs is to improve the defect of SDWGW since the color standard deviation is too small in the image. Max white method is better applied in the image with white point and better adjustment is exclusively applied to white color. Gray world assumption, standard deviation weighted gray world, and proposed fuzzy system white balance algorithm all forsake some quality of white color patch and focus on the quality of other colors in the whole image.

4. Conclusions

In this paper, a color correction system was proposed. The first part of the system is color cast detection, and the second is white balance algorithm. By using trained back-propagation neural network, the test images can be classified as having no cast, real cast, or intrinsic cast. An image with real cast must be fed into white balance algorithm to adjust colors, while an image without cast or with intrinsic cast doesn't need to be corrected.

Because of nonlinear classification by neural network, our methodology yields a m uch higher accuracy for cast detection than its alternatives. The proposed white balance algorithm is structured by the fuzzy system that has two inputs and one output, with the luminance effect in the input image take into account. Traditional white balance algorithms such as gray world assumption and max white method are restricted under some tough conditions. For the images with less number of colors or small standard deviation of colors, the proposed white balance algorithm can improve the quality of color correction.

Cast detection using neural network increases the accuracy of cast classification. However, it is extensively computational. The proposed white balance algorithm combines the luminance weighting and has better performance when the input images have less number of colors or small standard deviation of colors. If the input image has strong cast, the performance of the proposed white balance algorithm will not be satisfied. Generally speaking, white balance algorithms obtain the global area color correction weighting. Thus, white balance algorithm should obtain the local area weighting for the selective color correction of specific areas, such as skin, or sky, in order to reproduce what is held to be the genuine color.

References

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

[2] K. Barnard, L. Martin, A. Coath, B. Funt, A comparison of computational color constancy algorithms. II. Experiments with image data, IEEE Trans. Image Processing, 11 (9) (2002) 985-996.

[3] K. Barnard, V. Cardei, B. Funt, Estimating the Scene Illumination Chromaticity using a Neural Network, J. Opt. Soc. Amer. A, 19 (12) (2002) 2374-2386.

[4] G.D. Finlayson, S.D. Hordley, P.M. Hubel, Color by correlation: a simple, unifying framework for color constancy, IEEE Trans. Pattern Anal. Machine Intell. 23 (11) (2001) 1209-1221.

[5] G. Sapiro, Color and illuminant voting, IEEE Trans. Pattern Anal. Machine Intell. 21 (11) (1999) 1210-1215.

[6] F. Li, H. Jin, An approach of detecting image color cast based on image semantic, in Proc. of IEEE Conf.: Machine Learning and Cybernetics, pp. 3932-3936, 2004.

[7] F. Gasparini, R. Schettini, Color correction for digital photographs, in Proc. of IEEE Conference on Image Analysis and Processing, pp. 646 - 651, 2003.

[8] F. Gasparini, R. Schettini, P. Gallina, An Innovative Algorithm for Case Detection, in Proc. of SPIE, vol. 4672, pp. 280-286, 2002.

[9] F. Gasparini, R. Schettini, P. Gallina, Tunable Cast Remover for Digital Photographs, in Proc. of SPIE, vol. 5008, pp. 92-100, 2003.

[10] M. Friedddman, A. Kandel, Introduction to Pattern Recognition, Imperial College Press, London, 1999.