

A NOVEL APPROACH TO COLOR NORMALIZATION USING NEURAL NETWORK

H.D. Cheng, Xiaopeng Cai and Rui Min

The Department of Computer Science
Utah State University

ABSTRACT

Color is a powerful descriptor that often simplifies object extraction and identification, and many computer vision systems use color to aid object recognition. However, image colors strongly depend on lighting geometry (direction and intensity of light source) and illuminant color (spectral power distribution). Either small variation in the intensity or the change of scene illumination can dramatically make object color changed. To overcome the lighting dependency problem, a color constancy or normalization algorithm should be used for pre-processing.

This paper presents a novel approach to performing color normalization. A nonlinear mapping function is estimated using a neural network. Once the mapping function is found accurately, an image under unknown illumination may be transformed to the image under the predetermined illumination, which will be useful for color image processing. Three groups of experiments were conducted. In our experiments, images are processed by various neural networks and the performance is boosted by using a committee machine, and then the mapping errors is estimated and the results are compared with those of other algorithms. The experimental results demonstrate that the performance of the proposed method is superior to that of other color normalization algorithms.

Index Terms-- Color normalization, color constancy, neural networks, committee machine, genetic algorithm

I.INTRODUCTION

According to color theory [1, 2], the surface color of an object is partially determined by its surface reflectance and the spectral power distribution of the light. The color of objects may change if the illuminant changes. Figure 1 shows the colors of two images of the same ball that are very different under different illuminates. It is clearly to notice that the pink color in image (a) becomes the purple color in image (b). The color shift could cause problems in designing a computational vision system. To overcome the lighting dependency problem, a color constancy or normalization algorithm should be used as a pre-process.

In many articles, color constancy and color normalization are considered the same. However, there are some differences [3]. Usually, color constancy can be defined as the maintenance of color appearance despite the variations in the color of nearby objects and in the spectral power distribution of the ambient light [4]. The process of color constancy is usually divided into two steps, 1) estimate illuminant parameters; 2) utilize the parameters to compute illumination independent surface descriptors [5]. Four kinds of commonly used color constancy algorithms were summarized in [5, 6]: 1) Gray world methods; 2) Illuminant estimation by the maximum of each channel; 3) Gamut mapping methods; and 4) Color by correlation. The experimental results of the algorithms were also given in [5, 6]. Some color constancy algorithms attempt to estimate the spectral power distribution of the light source and the spectral reflectance functions of the surfaces in the scene [7-10]. After the illuminant color is estimated, image colors may be corrected and color bias may

be removed. Other prominent constancy algorithms are the grey world algorithm [11], retinex method [12], gamut mapping algorithm [13-16], and statistical color constancy [17-21].

The main purpose of the illuminant-invariant color normalization algorithms is to develop a good mapping function that transforms the object's colors under an unknown illuminant into a 'normalized' form. In gray level images, the mapping is fairly simple. We only need to consider the changing gray-levels influenced by the brightness. There are two classes of pixel brightness transformation (normalization): brightness corrections and gray-scale transformations [22]. Brightness correction modifies the pixel brightness, taking into account of its original brightness and its position in the image. Gray scale transformations change brightness without considering the position of the pixel in the image. There are many different color spaces to represent an image [23]. In RGB color space, the components are dependent each other, therefore, the color normalization becomes quite complex. [3, 24] developed an algorithm called comprehensive normalization that can reduce the variation due to illuminant color and lighting geometry. Let $(l_1\alpha_1, l_1\beta_{g_1}, l_1\gamma_{b_1})$ and $(l_2\alpha_2, l_2\beta_{g_2}, l_2\gamma_{b_2})$ denote image colors of two pixels where (l_1, l_2) and (α, β, γ) are scalars that model lighting geometry and illuminant color, respectively. Then the pixels become

$$\left\{ \begin{array}{l} \left(\frac{\alpha_1}{\alpha_1 + \beta_{g_1} + \gamma_{b_1}}, \frac{\beta_{g_1}}{\alpha_1 + \beta_{g_1} + \gamma_{b_1}}, \frac{\gamma_{b_1}}{\alpha_1 + \beta_{g_1} + \gamma_{b_1}} \right) \\ \left(\frac{\alpha_2}{\alpha_2 + \beta_{g_2} + \gamma_{b_2}}, \frac{\beta_{g_2}}{\alpha_2 + \beta_{g_2} + \gamma_{b_2}}, \frac{\gamma_{b_2}}{\alpha_2 + \beta_{g_2} + \gamma_{b_2}} \right) \end{array} \right. \quad (1)$$

and

$$\left\{ \begin{array}{l} \left(\frac{2l_1 r_1}{l_1 r_1 + l_2 r_2}, \frac{2l_1 g_1}{l_1 g_1 + l_2 g_2}, \frac{2l_1 b_1}{l_1 b_1 + l_2 b_2} \right) \\ \left(\frac{2l_2 r_2}{l_1 r_1 + l_2 r_2}, \frac{2l_2 g_2}{l_1 g_1 + l_2 g_2}, \frac{2l_2 b_2}{l_1 b_1 + l_2 b_2} \right) \end{array} \right. \quad (2)$$

under lighting geometry normalization and illuminant color normalization, respectively. The method in [3, 24] uses two normalizations iteratively until each normalization step is idempotent. This process always converges and the convergent image is unique. [25] presented an eigencolor normalization algorithm, which is based on the moments of color distributions and the algorithm is for planar patterns. This normalization method makes the color distribution less correlated and more compact, and the computed compact color image is transformed into a new color space by rotating the histogram to the reference axis. [26] developed a neural network to simulate the retinex algorithm. A neural network is trained to estimate the chromaticity of the illuminant in a scene [27, 28]. A linear technique (ridge regression) has been applied to color constancy and its result has been compared with that of neural network, support vector and gray world algorithm, and their comparisons are presented [32]. In this paper, a neural network is designed to perform the nonlinear mapping from an image under an illumination to the image under another illumination. The inputs and outputs are the intensities of the three color components in RGB color space.

II. PROPOSED METHOD

The proposed method may be applied to different color spaces for color normalization, for

example, RGB, HSV, etc. RGB color space is the most commonly employed and it can be converted into other color spaces easily, therefore, only RGB color space is discussed in this paper. However, the idea of this paper can be applied to other color spaces as well.

Let $\Omega = \{(r, g, b)^T \in \mathbb{R}^3 : 0 \leq r, g, b \leq 255\}$ be the set of image color vectors. Let the color value of an image pixel p at position (i, j) under the illuminant I_k be denoted by $c(p_{i,j}^k) \in \Omega, 1 \leq i \leq M, 1 \leq j \leq N$; where $M \times N$ is the size of the image. Suppose we have two color images of the same scene taken under different illuminants I_1 and I_2 . The image under I_2 is registered as the standard. Each pair of corresponding image pixels $p_{i,j}^1$ and $p_{i,j}^2$ in the two images represents a mapping: $\phi(I_1, I_2): c(p_{i,j}^1) \mapsto c(p_{i,j}^2), 1 \leq i \leq M, 1 \leq j \leq N$

The mapping can be represented as a 3×3 matrix A with the following equation:

$$c(p_{i,j}^2) = A \cdot c(p_{i,j}^1), \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (3)$$

For the linear methods, the elements of matrix A are constants. However, in the real world, the mapping $\phi(I_1, I_2)$ should be nonlinear. That is, the elements of matrix A are variables related with I_1 and I_2 . The nonlinear mapping $\phi(I_1, I_2)$ is not easy to be expressed explicitly. We use a neural network to perform such mapping. Once the neural network is trained, it can be consider as a “converter” transforming the images under one illuminant to the images under another illuminant, shown in Figure 1. Figure 2 displays the data in the RGB space of a scene taken under two different illuminants. If Figure 2 is registered as the standard image, the color of a point in Figure 1 should be mapped to the same color of the corresponding point in Figure 2 by the trained neural network.

A major task for the proposal algorithm is to design a good neural network according to the color characteristics. The following factors need to be considered:

1. Data selection and normalization.
2. The neural network architecture.
3. Training algorithm determination.
4. Boosting performance by committee machine method.

We use RGB vectors of image pixels as a training set. The number of colors in RGB space is 16,777,216. If all colors are considered, the system will be very sensitive to noise, and the processing time will be higher. Therefore, it is a good way to choose the pixels having different colors from the model image database. The “same” color is defined by the nearest m values. A small value of m may increase computational time greatly, if m increases k times, then the computational time will increase k^3 times. A large value of m may introduce unnecessary noise and distort the colors. Based on the experiments, $m=4$ is a suitable value. We divide each component of RGB space with $n=256/m=64$ bins, we can get at most $n^3=262,144$ different combinations in the whole training set. For example, there are 12682 different colors, based on $m=4$, in the database under the illuminant of Sylvania 75W halogen bulb. A number of pixels (N) are chosen as the training data. Each pixel per color is chosen as the training data in our experiments, so $N=12682$. The other pixels are used for evaluation. The number of test data is about 200 times of the training data. After selecting the training data, it needs to normalize the input data by dividing the color component with the maximum value 255. That is,

$$R' = R/255, G' = G/255, B' = B/255. R', G', B' \in [0,1]$$

We use a multilayer, feed-forward, error back-propagation neural network with one hidden layer. Each of the colors is an input vector of the neural network, and the output of the neural network is a vector of normalized color. Figure 3 shows the structure of a multilayer neural network and its parameters with their corresponding size. For example, W_i , $i=1,2$ are the $n \times 3$ matrix weight and θ_i , $i=1,2$ are the 3×1 vector bias. The input layer consists of three inputs, i.e., $X = c(p_{i,j}^1)$. The hidden layer consists of 100-1000 neurons. The output layer consists of three neurons, corresponding to the desired three components under another illuminant. The neurons in the hidden layer have the same sigmoid activation function. The neurons in the output layer have the linear activation function. For the multilayer neural networks, the outputs of one layer become the inputs of the next layer:

$$Y_0 = X; \quad Y_m = \varphi_m(W_m Y_{m-1} + \theta_m), m=1,2 \quad (4)$$

The next step is to propagate the error backward through the network:

$$\delta_2 = -2\Phi_2'(V_2)(Y - Y_2) \quad (5)$$

$$\delta_1 = \Phi_1'(V_1)W_2^T \delta_2 \quad (6)$$

where

$$\Phi_m'(V_m) = \begin{bmatrix} \frac{\partial \varphi_m(v_{1m})}{\partial v_{1m}} & 0 & \dots & 0 \\ 0 & \frac{\partial \varphi_m(v_{2m})}{\partial v_{2m}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \varphi_m(v_{nm})}{\partial v_{nm}} \end{bmatrix}, m=2,1 \quad (7)$$

$Y = c(p_{i,j}^2)$ is the target value, v_{im} , $i=1,2,\dots,n$ are members of V_m , $m=1,2$, i.e.,

$$V_m = [v_{1m}, v_{2m}, \dots, v_{mm}]^T, m = 2,1 \quad (8)$$

In experiments, the training algorithm updates weight and bias values at iteration k according to gradient descent momentum shown in Eq. (9) and Eq. (10):

$$\Delta W_m(k) = \gamma \Delta W_m(k-1) - (1-\gamma)\eta \delta_m Y_{m-1}^T, \quad m = 2,1 \quad (9)$$

$$\Delta \theta_m(k) = \gamma \Delta \theta_m(k-1) - (1-\gamma)\eta \delta_m, \quad m = 2,1 \quad (10)$$

where γ is the momentum coefficient, which is set as 0.9, and η is the adaptive learning rate. If the squared error over the entire training set increases by more than a certain percentage ξ (set $\xi=0.01$) after the weight update, the weight update is canceled; η is multiplied by a decreasing ratio 0.7; and γ is set to zero. If the squared error increases less than ξ after the weight update, the weight update is accepted; η is unchanged; and γ is set back to 0.9. If the squared error decreases after weight update, the weight update is accepted; η is multiplied by an increasing ratio 1.05; and γ is set back to 0.9. Momentum allows a neural network to respond not only to the local gradient, but also to the recent trends in the error surface. Acting like a low pass filter, momentum allows the neural network to ignore small features in the error surface. Without momentum a network may get stuck in a shallow local minimum [29].

In training, the condition for the termination of training is one of the follows and their corresponding parameters are listed in Table I.

1. The maximum number of epochs is reached.
2. The maximum amount of time has been exceeded.
3. Performance has been minimized to the goal.

4. The performance gradient falls below the minimum performance gradient.

In the experiments, a committee machine consists of two trained neural networks with the common inputs and their outputs are combined to produce an overall output by an ensemble averaging method, shown in Figure 4. $y_i = [r_i, g_i, b_i], i = 1, 2$ are the outputs of the two neural networks; and $y = [r_c, g_c, b_c]$ is the overall output of the committee machine. Let matrix K denote the weights of the combiner. We have the following relation:

$$[r_c, g_c, b_c]^T = K \cdot [r_1, g_1, b_1, r_2, g_2, b_2]^T \quad (11)$$

where K is a 3×6 matrix.

The key issue is to find an optimized matrix K . A genetic algorithm is used to search the optimized K . Let $a_{ij} \in [0,1], i = 1,2,3; j = 1,2, \dots, 6$ be the elements of matrix K representing the weights. We use 10 bits for each weight a_{ij} to guarantee the accuracy and use a large volume of initial population, which is 1000. Then, a chromosome with 180(=10*3*6) bits represents the whole matrix K , i.e., all the desired weights. The fitness function is to estimate the RMS error between the registered image and the result image using different matrix K . The steps of the genetic algorithm for training the weights of the committee machine are as follows,

1. Generate the initial population with 180 bits length chromosomes
2. Compute the fitness for each chromosome
3. Go to step 8 if the RMS error is lower than a threshold (ϵ) or the iteration time is greater than a threshold (T)
4. Pair chromosomes for chromosome mate
5. Process single point crossover
6. Do mutation on each chromosome

7. Go to step 2
8. Select the best chromosome

By using a committee machine, the expectation is that the differently trained machines converge to different local minima on the error surface, and the overall performance is improved by combining the outputs in some way [29]. Therefore, the overall nonlinear mapping can be presented by $\psi = K \cdot [\phi_1 \ \phi_2]^T$. Through this nonlinear mapping ψ , the pixels of an image under illuminant I_1 can be transformed to the corresponding pixels of the registered image under illuminant I_2 as follows:

$$p_{i,j}^2 = \psi(p_{i,j}^1) = K \cdot [\phi_1(p_{i,j}^1) \ \phi_2(p_{i,j}^1)]^T \quad (12)$$

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

To evaluate the performance of the proposed approach, color histogram distance and RMS in chromaticity are calculated and compared with those in [25] and [30]. We also show the results influenced by various neural network configurations. Finally, we boost the performance by using a committee machine.

The images were taken from an image database used by many researchers [3, 25, 30]. All images were collected using a Sony DXC-930 digital video camera, with 8 bits per channel dynamic range. Each image was captured under five different illuminants. The illuminants are Macbeth 5000K fluorescent tubes, Macbeth 5000 tubes with Roscolux #3202 full blue filter, Sylvania cool white fluorescent tube, Phillips Ultralume fluorescent tube, and Sylvania 75W halogen bulb. The database has images of 11 objects in a fixed orientation captured under 5 different illuminants. In total, there are 55 images [30]. All the utilized color images can be downloaded from

A. Errors and Comparisons

Figure 5 shows the original images “tide” under 5 different illuminants (the first and second row) and their corresponding transformed images (the last row). Figure 6 shows another example. We take the images under the illuminant “Macbeth 5000K fluorescent tubes” as the registered target images. The images under the other four illuminants are mapped to the registered target images. From the figures, the colors in all the mapped images are almost the same with the registered target images, and it is difficult to distinguish the differences visually.

Table II shows the mapping errors with different error metrics such as MAE (mean absolute error), RMS (root mean square error), μ (mean), σ (variance), D_H (histogram distance), ξ (the percentage of pixels with errors less than 8). The histogram distance is defined as:

$$D_H = \frac{\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \left| \frac{M(i, j, k)}{N_M} - \frac{T(i, j, k)}{N_T} \right|}{3} \quad (13)$$

where M and T are the color histograms of a mapped image and the target image with n bins for each color component, respectively ($n = 16$ in our experiments). $M(i, j, k)$ and $T(i, j, k)$ are the number of pixels in each bin of the respective histograms, and N_M and N_T are the total number of pixels in the mapped and target images, respectively.

Histogram distance could be used as a distance function for the color object recognition and color image retrieval [31]. [25] introduced an eigencolor normalization method for illumination-invariant color object recognition. Color histogram distance was used as the match function. The better match corresponds to the smaller distance between the test

image and the target image. Since the colors in the background of the most images are relative uniform and the colors in the region of interest (ROI) are diversiform, we focus on the color changing of ROI. [25] used 40×40 window in ROI; we also calculate the histogram distance at the same clipping level on the ROI and list the comparisons in Table III and Figure 7. From the table, the smaller the values are, the better the results. The results of our proposed approach are better than those obtained by the method in [25]. There is only one value “Javex_syl_cwf” smaller than the one using the proposed approach. However, it should be noticed that this value is much smaller than other values in [25]. In addition,, the results of the proposed approach are very close and consistent. The largest variation equals 0.2513 (0.2996 minus 0.3483) while the largest variation in [25] is 0.97 (1.3200 minus 0.3500).

The performance of color constancy or color normalization algorithms has been evaluated in terms of RMS difference in chromaticity between the mapped image and registered target image. In Table III, we calculate the RMS error with the three component values whose ranges are from 0 to 255. In order to compare our approach with other methods, we transform our results presented in the RGB space into values in the rg chromaticity space.

The data in the rg chromaticity space can be represented as:

$$\begin{cases} r = R/(R+B+G) \\ g = G/(R+B+G) \end{cases} \quad (14)$$

It is clear that the blue chromaticity component can be recovered from $b=1-r-g$, however, the absolute scene brightness information is lost.

Table IV lists the errors of six algorithms that are white-patch retinex, greyworld, 2D gamut-constraint, 3D gamut-constraint, neural net [30] and the neural network designed in this paper. “Nothing” denotes doing no color normalization (or color constancy). The error is the RMS difference in chromaticity between mapped image and the registered target image. The results are the averages of 220 tests, mapping for 55 images of 4 illuminants (Macbeth 5000 tubes with Roscolux #3202 full blue filter, Sylvania cool white fluorescent tube, Phillips Ultralume fluorescent tube, and Sylvania 75W halogen bulb). The results of the first six rows are taken from [30]. The last one is from the proposed approach. As shown in Table IV, the performance of the proposed method is the best.

In the proposed system, only one mapping function is needed for an unregistered illumination. Although the neural network training process is time-consuming, the mapping process using the trained neural network is extremely fast.

B. Results Influenced by Neural Network Configurations

Since the sizes of the input layer and output layer are fixed, we only consider the size of the hidden layer. The goal of the experiments is to determine the influence of the hidden layer size on the accuracy of the illuminant estimations. The neural networks with the hidden layer size from 100 to 1000 were tested. All the networks were trained with the same training data for 2000 epochs to assure good learning and network stability. Table V shows the average estimation errors and the best performance (the smallest estimation error) of each neural network trained 10 times. We use RMS error in chromaticity as the performance index. The networks with hidden layer size 400-600 size are better. However,

the differences among the performances of the networks are quite small. It shows that the hidden layer size has no much influence on the performance.

We also test if the neighbor pixels would influence the training performance of the neural networks. The original input data are taken by the local averaging. It is equivalent with smoothing the image. Table VI shows the performance of the neural networks with different window sizes. The input data is the average value of the pixels in an $N \times N$ window. We choose neural network “NN_200” in the experiment. In order to reduce the influence of the initialization, we train the network 10 times and take the average of results. From the table, we can conclude that the window size has little influence on the performance as well.

C. Performance Boosting by Committee Machine Method

In the experiments, we design two types of committees. The individual machines are back-propagation neural networks with hidden layer size 400 and hidden layer size 600, respectively. The training set is the same. The test set is also the entire database. The first type of committee simply averages the outputs of the two neural networks:

$$P_{i,j}^2 = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & .5 & 0 & 0 & .5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \cdot P_{i,j}^1 \quad (15)$$

where ϕ_1 and ϕ_2 are the neural networks whose hidden layer size are 400 and 600, respectively. The second type of committee is a weighted average of the outputs of the individual neural network. We use a genetic algorithm to train the weights. An example of the weights matrix is as below:

$$P_{i,j}^2 = \begin{bmatrix} 0.4377 & 0.0173 & 0.0491 & 0.4504 & 0.0724 & 0.0167 \\ 0.0382 & 0.4580 & 0.0550 & 0.0094 & 0.4488 & 0.0501 \\ 0.0219 & 0.0201 & 0.4699 & 0.0746 & 0.0434 & 0.4437 \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \cdot P_{i,j}^1 \quad (16)$$

Table VII shows MAE errors, RMS errors for the grey-level value (0-255), and RMS errors in chromaticity of these two types of committees and their individual neural networks. We transform images under the four illuminants (except images under the illuminant “Macbeth 5000K fluorescent tubes” that are registered target images) by the proposed method. The errors between two images are calculated. In total, there are 220 results. The results in Table VII are the averages of 220 results, estimated for 55 scenes taken under 4 illuminants. It is clearly shown that both the average committee and the genetic weighted committee can boost the performance.

D. Comparison by Content-based Image Retrieval

In order to investigate whether the proposed method is effective in the real application, the content-based image retrieval by color indexing is implemented in our experiments. The results by color indexing [31] with and without the proposed algorithm as the preprocessing step are compared. Because the result by color indexing was poor [3], we choose 24 out of 55 images in our database, which are five objects with largely different colors under four illuminations. The five objects are coffee, flowers, macaroni, shampoo and tide, and the four illuminations are halogen, Macbeth 5000 K, Phillips ultralume, and Sylvania cool white. Table VIII lists the comparison of results without the preprocessing and those with the preprocessing for content-based image retrieval. We compute the average retrieval accuracies for correct images in the first one, two and three. Each image in the 24 images is

used to query other 23 images. The results show the query accuracies are improved, especially the hitting ratio with the preprocessing of the first image is much higher than that without the preprocessing.

IV. CONCLUSIONS

Color is one of the prominent features for color object recognition. However, the same scene under different illuminants can result in different color images, and that will cause the difficulty in color object recognition. In this paper, we proposed a novel neural network approach for illuminant invariant color normalization to remove the color dependency due to the illumination. The proposed approach can effectively transform a color image under one illuminant to the corresponding image under the desired illuminant. The experimental results demonstrate that the proposed method outperforms the existing algorithms. It can improve the accuracy of color object recognition, image retrieval, etc., and find wide application in the areas related to color image processing.

REFERENCES

- [1] K. Nassau, *Color for science, art and technology*. Amsterdam: Elsevier, 1998.
- [2] L. W. MacDonald and M. R. Luo, *Colour image science: exploiting digital media*. Chichester, England: Wiley, 2002.
- [3] G. D. Finlayson and G. Y. Tian, "Color normalization for color object recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, pp. 1271-1285, 1999.
- [4] D. Jameson and L. M. Hurvich, "Essay Concerning Color Constancy," *Annual Review of Psychology*, vol. 40, pp. 1-22, 1989.
- [5] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms - Part I: Methodology and experiments with synthesized

- data," *IEEE Transactions on Image Processing*, vol. 11, pp. 972, 2002.
- [6] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms - Part II: Experiments with image data," *IEEE Transactions on Image Processing*, vol. 11, pp. 985, 2002.
 - [7] L. T. Maloney and B. A. Wandell, "Color Constancy - a Method for Recovering Surface Spectral Reflectance," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 3, pp. 29-33, 1986.
 - [8] M. Dzmura, "Color Constancy - Surface Color from Changing Illumination," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 9, pp. 490-493, 1992.
 - [9] S. Tominaga and B. A. Wandell, "Component Estimation of Surface Spectral Reflectance," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 7, pp. 312-317, 1990.
 - [10] B. J. Craven and D. H. Foster, "An Operational Approach to Color Constancy," *Vision Research*, vol. 32, pp. 1359-1366, 1992.
 - [11] G. Buchsbaum, "A spatial processor model for object color perception," *Journal of the Franklin Institute*, vol. 310, pp. 1-26, 1980.
 - [12] E. H. Land, "The Retinex theory of color vision," *Scientific American*, vol. 237, pp. 108-129, 1977.
 - [13] D. A. Forsyth, "A Novel Algorithm for Color Constancy," *International Journal of Computer Vision*, vol. 5, pp. 5-36, 1990.
 - [14] K. Barnard, "Improvements to gamut mapping colour constancy algorithms," *Computer Vision - Eccv 2000, Pt I, Proceedings*, vol. 1842, pp. 390-403, 2000.
 - [15] S. Tominaga, S. Ebisui, and B. A. Wandell, "Scene illuminant classification: Brighter is better," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 18, pp. 55-64, 2001.
 - [16] S. Tominaga and B. A. Wandell, "Natural scene-illuminant estimation using the sensor correlation," *Proceedings of the IEEE*, vol. 90, pp. 42-56, 2002.
 - [17] D. H. Brainard and W. T. Freeman, "Bayesian color constancy," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 14, pp. 1393-1411, 1997.
 - [18] K. Barnard, L. Martin, and B. Funt, "Color by correlation in a three dimensional color space," presented at 6th European Conference on Computer Vision, Dublin, Ireland, 2000.
 - [19] 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, pp. 1209, 2001.
 - [20] C. Rosenberg, M. Hebert, and S. Thrun, "Color constancy using KL-divergence," Vancouver, BC, 2001.
 - [21] J. M. Geusebroek, R. Van den Boomgaard, A. W. M. Smeulders, and H. Geerts, "Color invariance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1338, 2001.

- [22] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*, 2nd ed. Pacific Grove, Calif.: PWS Pub., 1999.
- [23] H. D. Cheng, X. H. Jiang, Y. Sun, and J. L. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition*, vol. 34, pp. 2259-2281, 2001.
- [24] G. D. Finlayson, B. Schiele, and J. L. Crowley, "Comprehensive color image normalization," presented at Proceedings of the 5th European Conference on Computer Vision 1, Freiburg, Germany, 1998.
- [25] Z. Y. Lin, J. X. Wang, and K. K. Ma, "Using eigencolor normalization for illumination-invariant color object recognition," *Pattern Recognition*, vol. 35, pp. 2629-2642, 2002.
- [26] S. M. Courtney, L. H. Finkel, and G. Buchsbaum, "Network Simulations of Retinal and Cortical Contributions to Color Constancy," *Vision Research*, vol. 35, pp. 413-434, 1995.
- [27] B. Funt, V. C. Cardei, and K. Barnard, "Neural network color constancy and specularly reflecting surfaces," presented at Processing of AIC Color 97 II, Kyoto, Japan, 1997.
- [28] V. C. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity by using a neural network," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 19, pp. 2374-2386, 2002.
- [29] S. S. Haykin, *Neural networks: a comprehensive foundation*, 2nd ed. Upper Saddle River, N.J. New York: Prentice Hall: Institute of Electrical and Electronics Engineers, 1999.
- [30] B. Funt, K. Barnard, and L. Martin, "Is machine colour constancy good enough?" vol. 1406, *Lecture Notes in Computer Science*, 1998, pp. 445.
- [31] M. J. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, pp. 11-32, 1991.
- [32] V. Agarwal, A. V. Gribok and M. A. Abidi, "Machine learning approach to color constancy", *Neural Networks*, (20), pp. 559-563, 2007.

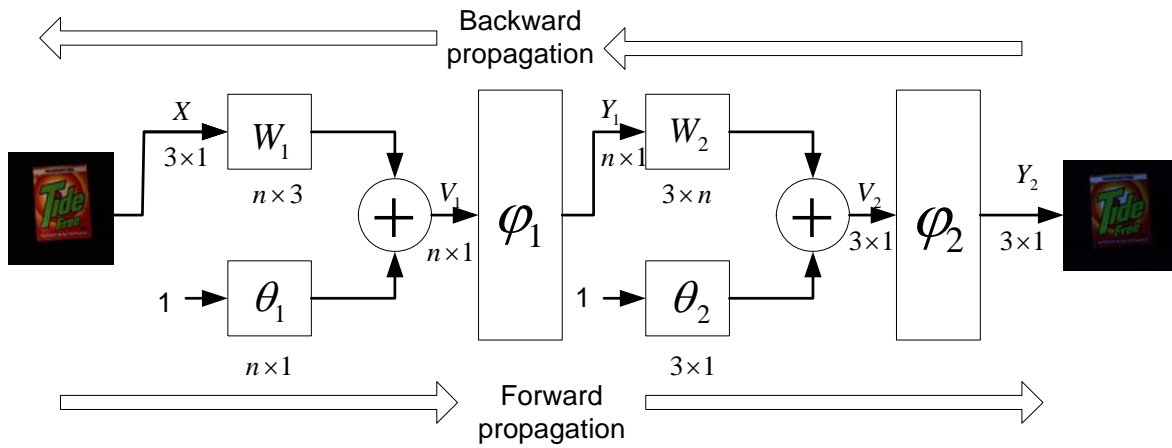


Fig. 3. The backpropagation neural network.

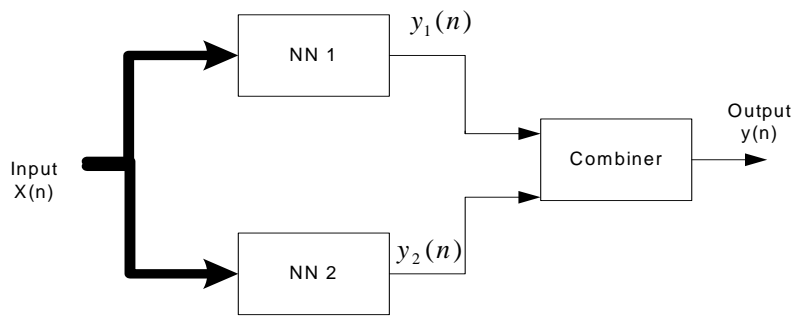
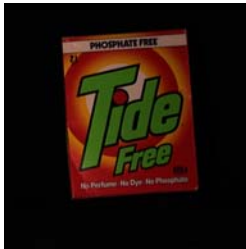


Fig. 4. Block diagram of a committee machine



(a)

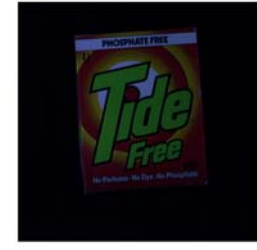


(b)

(c)

(d)

(e)



(f)

(g)

(h)

(i)

Fig. 5. “Tide” images: (a) the registered target image under the illuminant “Macbeth 5000K fluorescent tubes”, (b) (c) (d) (e) (the second row) images under the illuminants “Sylvania 75W halogen bulb”, “Macbeth 5000 tubes with Roscolux #3202 full blue filter”, “Phillips Ultralume fluorescent tube”, and “Sylvania cool white fluorescent tube”, respectively. (f) (g) (h) (i) (the last row) transformed images of the (b) (c) (d) (e), respectively.

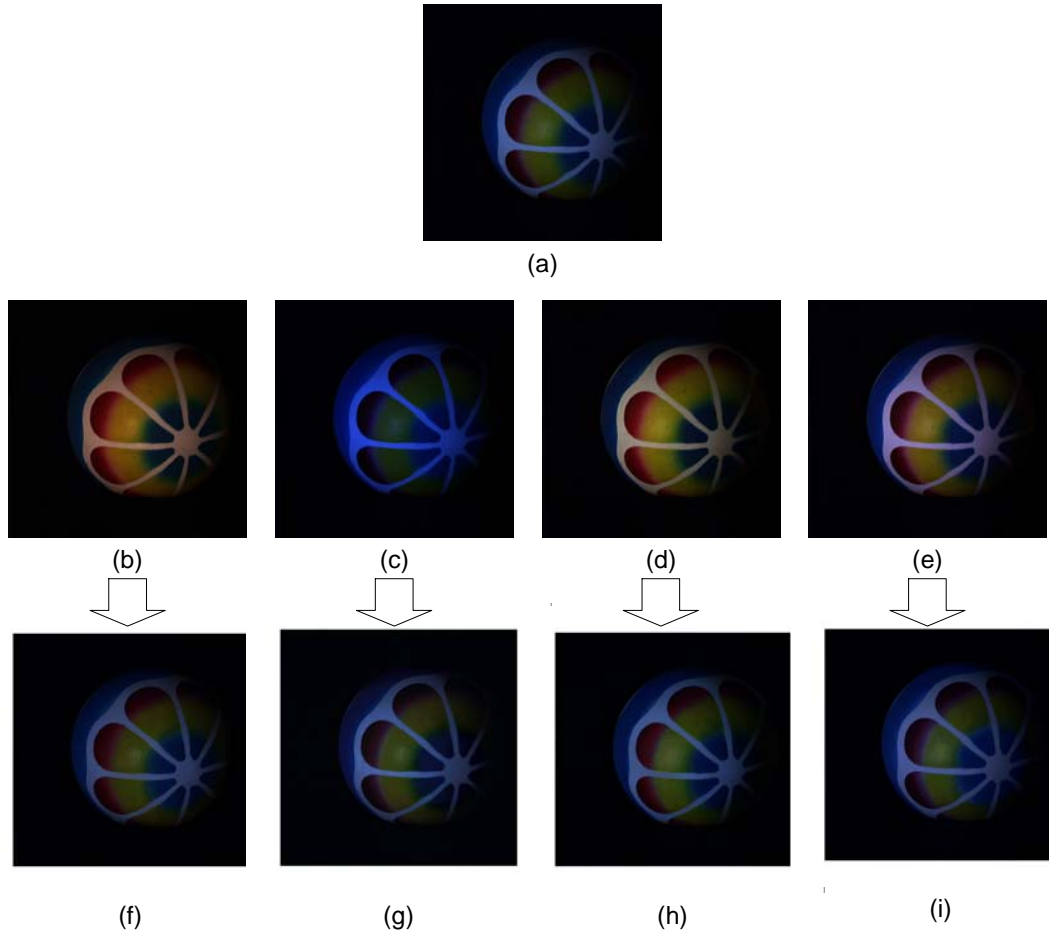


Fig. 6. “ball” images: (a) the registered target image under the illuminant “Macbeth 5000K fluorescent tubes”, (b) (c) (d) (e) (the second line) images under the illuminants “Sylvania 75W halogen bulb”, “Macbeth 5000 tubes with Roscolux #3202 full blue filter”, “Phillips Ultralume fluorescent tube”, and “Sylvania cool white fluorescent tube”, respectively. (f) (g) (h) (i) (the bottom line) transformed images of the images (b) (c) (d) (e), respectively.

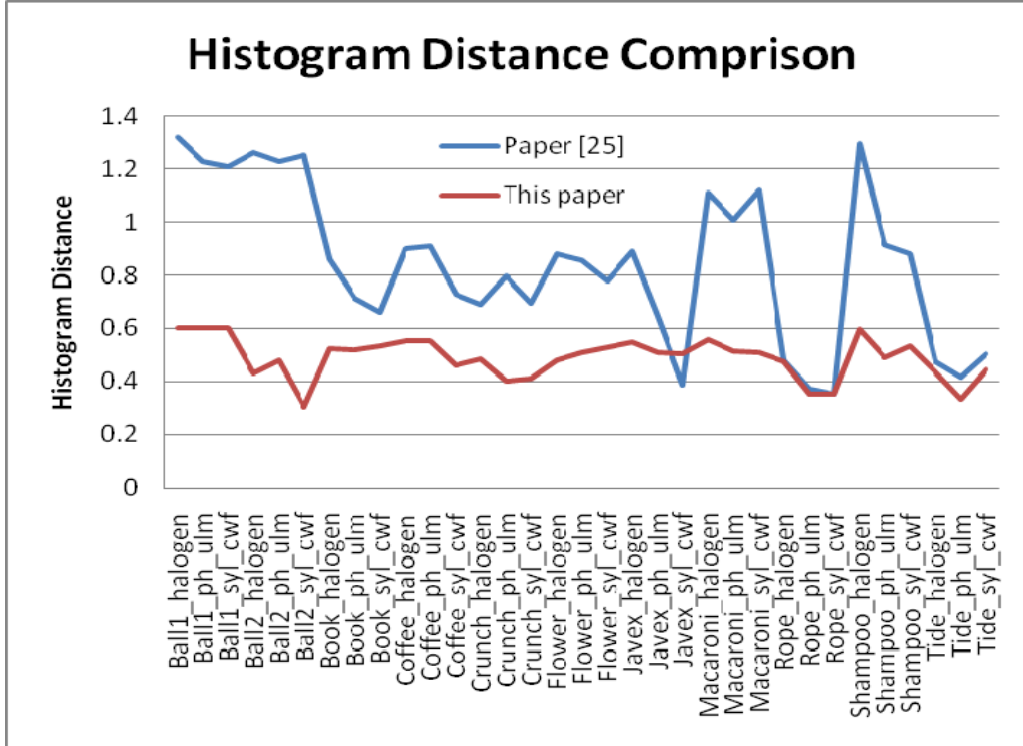


Fig. 7. Histogram distance comprison on each test image between this paper and [25].

Table I. Parameters used for training algorithm

Parameters	Value used	Remark
Net.trainParam.epochs	5000	Maximum number of epochs to train
Net.trainParam.goal	10	Performance goal
Net.trainParam.lr	0.01	Learning rate
Net.trainParam.lr_inc	1.05	Ratio to increase learning rate
Net.trainParam.lr_dec	0.7	Ratio to decrease learning rate
Net.trainParam.max_fail	10	Maximum validation failures
Net.trainParam.max_perf_inc	1.04	Maximum performance increase
Net.trainParam.mc	0.9	Momentum constant
Net.trainParam.min_grad	1e-5	Minimum performance gradient
Net.trainParam.show	50	Epochs between displays
Net.trainParam.time	Inf	Maximum time to train in seconds

Table II. The errors between the mapped image and the target image

Mapped Image	MAE	RMS	μ	σ	D_H	ξ
Ball1_halogen	2.7382	5.3308	0.5581	5.3015	0.2705	91.83%
Ball1_ph_ulm	1.8466	3.8839	-0.1804	3.8798	0.2117	94.91%
Ball1_syl_cwf	1.8851	3.7408	0.8807	3.6357	0.2485	95.73%
Ball1_mb+3202	3.1920	4.6393	1.9277	4.2199	0.2043	92.02%
Ball2_halogen	2.4186	5.0924	-0.2585	5.0858	0.1915	92.48%
Ball2_ph_ulm	1.6273	3.1789	0.2333	3.1703	0.1557	96.63%
Ball2_syl_cwf	2.1092	4.0044	0.9503	3.8900	0.2207	93.84%
Ball2_mb+3202	2.9324	4.3108	1.3979	4.0779	0.1548	95.40%
Book_halogen	2.5791	5.0632	1.9864	4.6572	0.2532	92.03%
Book_ph_ulm	1.8619	3.2914	-1.0889	3.1061	0.2786	96.09%
Book_syl_cwf	2.1810	3.6132	-0.0946	3.6120	0.3323	95.52%
Book_mb+3202	2.9107	4.4555	1.7884	4.0808	0.0389	96.11%
Coffee_halogen	1.3527	2.9255	0.4091	2.8968	0.0835	97.68%
Coffee_ph_ulm	1.0965	2.3616	0.8416	2.2066	0.0561	98.36%
Coffee_syl_cwf	1.3955	3.2011	1.2148	2.9617	0.1197	97.17%
Coffee_mb+3202	2.3121	3.2537	1.7871	2.7190	0.0276	97.99%
Crunch_halogen	2.0996	4.1308	1.6892	3.7696	0.2036	93.98%
Crunch_ph_ulm	1.3489	2.3138	0.8957	2.1334	0.1247	98.35%
Crunch_syl_cwf	1.7056	3.3018	1.2324	3.0631	0.1807	96.36%
Crunch_mb+3202	3.0149	4.2953	1.9207	3.8419	0.1818	94.50%
Flower_halogen	6.0914	11.5547	-0.5773	11.5403	0.3741	76.88%
Flower_ph_ulm	4.8914	9.6931	-0.5241	9.6789	0.3933	82.58%
Flower_syl_cwf	4.3462	7.9159	-0.3504	7.9081	0.3295	84.25%
Flower_mb+3202	6.1108	9.8070	-3.3915	9.2019	0.3588	77.90%
Javex_halogen	1.8637	4.2332	-0.3399	4.2195	0.1629	94.76%
Javex_ph_ulm	1.2285	2.3072	0.5402	2.2431	0.1147	98.08%
Javex_syl_cwf	1.3796	2.8626	1.0985	2.6435	0.1142	97.07%
Javex_mb+3202	2.6092	4.0437	1.5290	3.7435	0.0255	97.45%
Macaroni_halogen	1.5195	3.4798	0.8393	3.3771	0.1073	95.92%
Macaroni_ph_ulm	0.9902	2.0053	0.4092	1.9631	0.0668	98.64%
Macaroni_syl_cwf	1.6195	3.2500	1.2586	2.9964	0.1399	95.58%
Macaroni_mb+3202	2.8020	5.9263	1.2962	5.7828	0.0343	95.34%
Rope_halogen	2.9151	6.8797	-2.5433	6.3923	0.0895	94.49%
Rope_ph_ulm	3.0625	7.0636	-2.9317	6.4265	0.0598	94.02%
Rope_syl_cwf	2.4101	6.1754	-2.2721	5.7422	0.1054	95.49%
Rope_mb+3202	3.4950	8.3364	-2.5134	7.9485	0.1140	92.88%
Shampoo_halogen	1.5431	3.4269	0.5010	3.3901	0.0861	95.62%
Shampoo_ph_ulm	1.1630	2.3806	0.6492	2.2903	0.0745	97.84%
Shampoo_syl_cwf	1.2302	2.4829	0.9876	2.2780	0.0746	98.27%
Shampoo_mb+3202	2.9434	4.8827	2.0198	4.4454	0.0516	96.41%
Tide_halogen	1.8712	4.1095	0.4955	4.0795	0.1313	95.33%
Tide_ph_ulm	1.4815	2.7958	0.6294	2.7240	0.1351	97.98%
Tide_syl_cwf	1.7406	2.9682	0.6864	2.8877	0.1578	97.54%
Tide_mb+3202	3.4199	5.3188	1.7040	5.0384	0.2901	91.68%

Table III Histogram distance results compared with those in [25]

Images	Results*	Images	Results*	Images	Results*
Ball1_halogen	1.3200 0.5996	Crunch_halogen	0.6900 0.4825	Rope_halogen	0.4775 0.4725
Ball1_ph_ulm	1.2250 0.5996	Crunch_ph_ulm	0.7950 0.4008	Rope_ph_ulm	0.3700 0.3508
Ball1_syl_cwf	1.2050 0.5996	Crunch_syl_cwf	0.6950 0.4113	Rope_syl_cwf	0.3500 0.3483
Ball2_halogen	1.2625 0.4296	Flower_halogen	0.8775 0.4796	Shampoo_halogen	1.2950 0.5933
Ball2_ph_ulm	1.2275 0.4783	Flower_ph_ulm	0.8550 0.5071	Shampoo_ph_ulm	0.9150 0.4850
Ball2_syl_cwf	1.2500 0.3013	Flower_syl_cwf	0.7750 0.5275	Shampoo_syl_cwf	0.8800 0.5288
Book_halogen	0.8600 0.5229	Javex_halogen	0.8875 0.5454	Tide_halogen	0.4725 0.4354
Book_ph_ulm	0.7125 0.5167	Javex_ph_ulm	0.6450 0.5075	Tide_ph_ulm	0.4125 0.3304
Book_syl_cwf	0.6600 0.5313	Javex_syl_cwf	0.3850 0.5025	Tide_syl_cwf	0.5000 0.4442
Coffee_halogen	0.9000 0.5496	Macaroni_halogen	1.1100 0.5558		
Coffee_ph_ulm	0.9100 0.5521	Macaroni_ph_ulm	1.0075 0.5112		
Coffee_syl_cwf	0.7250 0.4592	Macaroni_syl_cwf	1.1200 0.5062		

*The values shown in boldface are the results of this paper. The smaller the value is, the better the result is.

Table IV RMS errors in chromaticity of six algorithms

Algorithm	RMS error
Nothing	0.1114
White-Patch Retinex	0.0625
Greyworld	0.0975
2D Gamut-Constraint	0.0649
3D Gamut-Constraint	0.0555
Neural Net	0.0643
The proposed method	0.0542

Table V. The performance of neural network with different size of hidden layer

nn	nn_100	nn_200	nn_300	nn_400	nn_500
Ave	0.0771	0.0724	0.0748	0.0706	0.0702
Best	0.0681	0.0598	0.0626	0.0542	0.0546
nn	nn_600	nn_700	nn_800	nn_900	nn_1000
Ave	0.0704	0.0760	0.0777	0.0774	0.0776
Best	0.0543	0.0671	0.0700	0.0724	0.0656

Table VI. The performance of the neural network with different window size

Size	3*3	5*5	7*7	9*9	11*11
RMS	0.0763	0.0763	0.0753	0.0755	0.0788
Size	13*13	15*15	17*17	19*19	21*21
RMS	0.0755	0.0764	0.0757	0.0715	0.0800
Size	23*23	25*25	27*27	29*29	31*31
RMS	0.0750	0.0759	0.0777	0.0728	0.0724
Size	33*33	35*35	37*37	39*39	41*41
RMS	0.0748	0.0728	0.0799	0.0773	0.0833

Table VII. Performance boosting by the committee machines

Algorithm	RMS_c	RMS	MAE
NN_400	0.0543	4.7436	3.1040
NN_600	0.0542	4.5613	3.0049
Average_committee	0.0539	4.5587	3.0021
Genetic_committee	0.0498	4.3685	2.9809

Table VIII. Comparison by Content-based Image Retrieval

	Without preprocessing	With preprocessing
1	40.0%	55.0%
2	35.0%	45.0%
3	33.3%	35.0%