A NOVEL APPROACH TO ILLUMINANT INVARIANT COLOR NORMALIZATION

H. D. Cheng and Xiaopeng Cai
Department of Computer Science
Utah State University, Logan, UT 84322-4205

ABSTRACT

This paper presents a novel approach to performing illuminant invariant color normalization. A neural network is used to conduct a nonlinear mapping of the images under one illuminant to the images under other illuminants. Once the mappings are found accurately, we can transform a color image under unknown illumination to the image under the predetermined illumination, which will be useful for color image processing. Three groups of experiments were conducted in this paper. We evaluate the mapping errors and compare them with those of other algorithms, present the results using various neural network configurations, and boost the performance by using committee machine. The experimental results demonstrate that the performance of our method is superior to other existing color constancy or color normalization algorithms.

1. INTRODUCTION

According to the 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. If the illuminant changes, the color will also change. To overcome the lighting dependency, a color constancy or normalization algorithm should be used as a pre-processing.

The main purpose of the most illumination-invariant color normalization algorithms is to develop a mapping that transforms the colors of an object under an unknown illumination into a 'normalized' form. In gray-level images, the mapping is much simpler. We only need to consider the changing gray-levels influenced by the brightness. In RGB color space, each pixel has three components RGB and these components are dependent on each other, therefore, the color normalization becomes more complex.

[3, 4] developed an algorithm called comprehensive normalization that can reduce the variation due to illuminant color and lighting geometry. The method uses these two normalizations iteratively until each normalization step is idempotent. This process always converges and the convergent image is unique. [5] presented a color image normalization method called eigencolor normalization, which was based on the foundation of the moments of color distributions and the normalization algorithm for planar patterns. This normalization method applied a compacting method to make the color distribution less correlated and more compact, and the computed compact color image was transformed into a new color space by rotating the histogram to the reference axis.

Neural network provides an alternative way for solving the color normalization problem. Previous neural network approaches to color normalization or color constancy fell into two categories. One used neural network to simulate some existing algorithms. For example, [6] developed a neural network to simulate the Retinex algorithm. The other was training a neural network to estimate the chromaticity of the illuminant in a scene [7]. In this paper, a neural network is designed to perform the nonlinear mapping from an image under one illumination to the image under another. The inputs and outputs are the intensities of the three color components in RGB color space.

2. PROPOSED METHOD

Let $\Omega = \{(r,g,b)^T \in \Re^3 : 0 \le r,g,b \le 255\}$ be the set of all image color vectors. Let the color value of an image pixel p at position (i, j) under the illuminant I_k ($k=1,2\ldots5$) be denoted by $c(p^k_{i,j}) \in \Omega$, $1 \le i \le M$, $1 \le j \le N$; $M \times N$ 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 $p^1_{i,j}$ and $p^2_{i,j}$ in the two images represents a mapping:
$\phi(I_1, I_2) : c(p^1_{i,j}) \mapsto c(p^2_{i,j}), 1 \le i \le M, 1 \le j \le N$.

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

$$c(p^2_{i,j}) = A \cdot c(p^1_{i,j}), \qquad 1 \le i \le M, 1 \le j \le N, (3)$$

For the linear methods, the elements of matrix A are constants. In this paper, we assume that mapping $\phi(I_1, I_2)$ is 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 neural network to do such mapping. Once the neural network is trained, it can be consider as a "converter" transforming images under one illuminant to images under another illuminant.

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

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

In order to generate the training data, it is a good way to choose the pixels having different colors from the model image database. With RGB color model, if we divide each component with n bins, we can get at most n^3 different colors. We consider two points with the same color if the difference of each component is less than 4, i.e., $\max(|r_1 - r_2|, |g_1 - g_2|, |b_1 - b_2|)$. For example, there are 12682 different colors in the database under the illuminant of Sylvania 75W halogen bulb. We choose 12682 pixels as the training data. The other pixels are for test evaluation. The number of test data is about 200 times of 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 to perform our experiments. 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 (*logsig* function in MATLAB). The neurons in the output layer have the same linear activation function (*pureline* function). In our experiments, the training algorithm updates weight and bias values according to gradient descent momentum and an adaptive learning rate. Momentum allows a network to respond not only to the local gradient but also to recent trends in the error surface. Acting like a low pass filter, momentum allows the network to ignore small features in the error surface. Without momentum, a network may get stuck in a shallow local minimum. With momentum, a network can slide through such a minimum.

We use the function *traingdx* in neural network toolbox of MATLAB to implement the training algorithm. The condition for the termination of training is one of the follows:

1. The maximum number of epochs is reached (5000)
2. The maximum time has been exceeded (infinite).
3. Performance has been minimized to the goal (10).
4. The performance gradient falls below min_grad (1e-5).

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

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

where K is a 3×6 matrix. A genetic algorithm is used to optimize this matrix. Let $a_{ij} \in [0,1], i = 1,2,3; j = 1,2,\cdots 6$ be the element of matrix K representing the weights. We use 10 bits for each weight to guarantee its accuracy and the large of initial population. The initial population is 1000. Then, a chromosome with 180 bits length represents the matrix K, i.e., all the weights we want. The fitness of the chromosome is the RMS error in chromaticity. For each generation, the best chromosome of the two parents and two children chromosomes is chosen and the other chromosome is randomly chosen by the roulette wheel selection method for mating. Then one single point crossover operation and one mutation operation are used on the chosen chromosomes. The mutation rate is set as 0.1. If the fitness of the best chromosome improves the performance of the individual machine 10%, the genetic algorithm is terminated.

Therefore, our overall nonlinear mapping can be presented by $\psi = K \cdot [\phi_1 \quad \phi_2]^T$. Through this nonlinear mapping ψ, a sequence of pixels of image under illuminant I_1 can be transformed to their corresponding pixels of the registered image under illuminant I_2 as follows:

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

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

To evaluate the performance of the proposed approach, color histograms distance and RMS in chromaticity are calculated and compared with other five algorithms. We also show the results influenced by various neural network configurations. Finally, we boost the performance by using committee machine.

The images were taken from an image database used by many researchers [3, 5, 8]. 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 [8].

3.1 Comparisons

Figure 1 shows the original images "tide" under 5 different illuminants (the first and second line) and their corresponding transformed images (the bottom line). We take 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. It is difficult to distinguish their differences with human eyes.

The performance of many color constancy or color normalization algorithms has been reported in terms of RMS difference in chromaticity between mapped image and registered target image. 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:

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

Table 1 lists the errors of six algorithms that are white-patch retinex, greyworld, 2D gamut-constraint, 3D gamut-constraint, neural net and the neural network designed in this paper. "Nothing" denotes doing no color normalization (or color constancy) [8]. 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 five rows are taken from [8]. The last one is from the proposed approach. As shown in Table 1, performance of the proposed method is the best.

3.2 Influenced of Network Configurations

Since the size of input layer and output layer are fixed that are both three color components. Here we only consider the size of hidden layer. The goal of the experiments is to determine the influence of the hidden layer size on the accuracy of the illuminant estimations. We test the neural networks with the size of hidden layer from 100 to 1000. All the networks were trained with the same training data and trained for 2000 epochs to assure good learning and network stability. Table 2 shows the average estimation errors and the best performance (the smallest estimation error) of each neural network trained 10 times. Here, we use RMS error in chromaticity as the performance index. The networks with 400-600 size hidden layer are better. However, the difference among the performances of the networks is small. It shows that the size of hidden layer has no big 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 local averaging. It is equivalent with smoothing the image. Table 3 shows the performance of neural network with different window size. The input data is the average value of pixels in a $N \times N$ window. We choose neural network "NN_200" in this experiment. Here "NN_200" means the hidden layer size is 200. In order to reduce the influence of initialization, we train the 10 times and have the average of them. From the table, we can conclude that the window size has little influence on the performance.

3.3 Performance Boosting by Committee Machines

In the experiments, we design two types of committees. The individual machines are back-propagation neural networks with 400 neurons in hidden layer and 600 in hidden layer, respectively. The training set is the same as the training set of individual neural network. The test set is also the entire database. The first type of committee simply averages the outputs of the two neural networks. The three output color components are averaged, respectively, shown as follows:

$$p_{i,j}^2 = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & .5 & 0 & 0 & .5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \cdot p_{i,j}^1$$

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 genetic algorithm to train the weights. An example of the weights matrix shows as follows:

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

Table 4 shows MAE errors, RMS errors for 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 4 are the average of 220 results, estimated for 55 scenes taken under to each of 4 illuminants. It is clearly shown that whatever the average committee and the genetic weighted committee can boost the performance.

4. CONCLUSTIONS

In this paper, neural network approach is applied for illuminant invariant color normalization. Our experiments show that the results of our method outperform the former algorithms'. With our approach, we can transform a color image under one illuminant to the corresponding image under another desired illuminant. It is very important to improve the performance of object recognition, and therefore, can improve the retrieval accuracy of some color-based image retrieval techniques.

5. REFERENCES

[1] K. Nassau, Color for science, art and technology, Amsterdam, New York: Elsevier, 1998.

[2] L. W. MacDonald and M. R. Luo, Colour image science: exploiting digital media, Hoboken, NJ: Wiley, 2002

[3] G. D. Finlayson, G. Y. Tian, Color normalization for color object recognition, Internat. J. Pattern Recogn. Artificial Intell. 13 (8) (1999) 1271-1285.

[4] G. D. Finlayson, B. Schiele, and J. L. Crowley, Comprehensive color image normalization, proceedings of the 5th European Conference on Computer Vision 1 (1998) 475-490, Freiburg, Germany.

[5] Z. Lin, J. Wang, and K. K. Ma, Using eigencolor normalization for illumination-invariant color object recognition", Pattern Recognition 35 (2002) 2629-2642.

[6] S. Courtney, L. H. Finkel and G. Buchsbaum, Network simulations of retinal and cortical contributions to color constancy, Vision Research 35 (3) (1995) 413-434.

[7] B. Funt, V. C. Cardei and K. Barnard, Neural network color constancy and specularly reflecting surfaces, Proc. AIC Color 97 II (1997) 523-526, Kyoto, Japan.

[8] B. Funt, K. Barnard, and L. Martin, Is machine colour constancy good enough, Proceedings of the 5th European Conference on Computer Vision, (1998) 446-459.

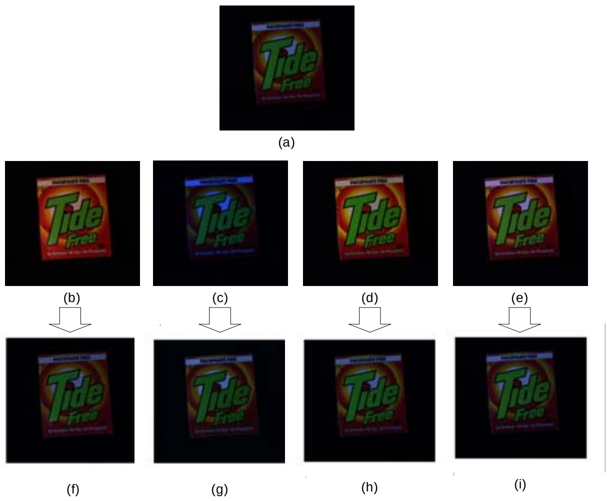

Figure 1. "Tide" images: (a) the registered target image under the illuminant "Macbeth 5000K fluorescent tubes", (b) (c) (d) (e) (the second line) images under the illuminants "Sylvania 75W halogen bulb", "Macbeth 5000 tubes with Roscolux #3202 full blue filter", "Phillips Ultralume fluorescent tube", and "Sylvania cool white fluorescent tube", respectively. (f) (g) (h) (i) (the bottom line) transformed images of the images (b) (c) (d) (e), respectively.

Table 4. Performance boosting by 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 1 RMS errors in chromaticity by 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 proposal method	0.0542

Table 2. 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 3. The performance of 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