A Global Color Transformation Scheme between Images Using Radial Basis Function Network Based on Discrete Wavelet Transform

Chen-Chung Liu
National Chin-Yi University of Technology,
Department of Electronic Engineering
ccl@ncut.edu.tw

Shih-Yao Jian
National Chin-Yi University of Technology,
Department of Electronic Engineering
fries49923@hotmail.com

Abstract

Color is the most important feature in color images and is applied in many different areas, such as medical image analysis, video object extraction, image compression, tracking systems, etc. Transferring a source image's color to a target image involves changes the target image's color to enhance the target image's color features. The color transfer from an image to another image can always enhance the color-transferred image's color features for wide applications in different fields. In this paper, a radial basis function network (RBFN) based on discrete wavelet transform algorithm for global color transferring between images is proposed. RBFN is trained on the R, G, and B planes to obtain the corresponding trained RBFN for each plane, respectively. The transferred intensity of each pixel on each plane of the target image is then evaluated using the trained RBFN. These transferred R, G, and B planes of the target image are then combined into the result color image result. The experimental results show that the proposed approach has three major advantages: (A) the proposed algorithm is manual free, simple, effective and accurate in transferring color between images without any change in the image details, (B) the proposed algorithm is time saving, due to operation directly and completely on RGB domain; it does not need the transformations among color spaces, (C) there are no restrictions in the image dynamic color ranges in the proposed algorithm, (D) the discrete wavelet transform can efficiently obtain feature pixels for training the RBFN.

Keywords: *color transfer, wavelet, RBFN.*

1. Introduction

Digital color images are the most important media for efficient information presentation and communication. They are widely applied in the multimedia, biomedical, Internet, and video and push people to pay more attention to color image processing to meet human requirements [1]. Color is a significant source of information for image analysis [2], classification [3], segmentation, recognition, and retrieval [4]. It provides artists with the ability to show creativity and style [5] and also offers doctors a means to describe tissue pathology and aesthetics in dermatology and dentistry [6]. Because some of the original colors in an image might not be suitable for analysis, the colors must be adjusted. The appearance of an image can be adjusted manually to obtain specific effects but that requires advanced image manipulation techniques which consume a great deal of time. Color transferring is one of the most popular techniques in image color processing for changing an image's color and preserving the image's original details and natural look at the same time. This means replacing the target image's colors using the source image's color without any change in the text and certain visual qualities [7].

Color transfer between images is a challenging task because both the source and target images may consist of various colors, causing multiple shades in the transfer results [8]. Several schemes have been proposed for global color transfer between images. Reinhard et al. [9] proposed a statistical based scheme. Their idea is based on transferring the original RGB color images into the $l\alpha\beta$ color space proposed by Ruderman et al. [10], in which the correlation between channels is minimized and the color perception can be evaluated better. They assumed that the pixel values of each channel in the $l\alpha\beta$ color space is a Gaussian distribution and the color transfer process shifts and scales each channel's pixel values from the target image to match the corresponding source image mean and variance because of the uncorrelated $l\alpha\beta$ color space axes. The color transferred $l\alpha\beta$ image is finally converted back into the RGB color space. Reinhard et al.'s scheme allows operating the transfer independently in each channel to convert a potentially complex three dimensional problem into three much simpler one dimensional problems. Although this technique is simple and efficient for a large range of images, it has two disadvantages; (A) it is time consuming because it must perform color space conversion three times to transfer color from a source image into a target image; (B) the color transfer quality is depends on the composition of the target and source images [5]; (C) overflow will appear when the image hue is oversaturated and may not be good for images with a big gap between different chromaticity's. S. Xu et al. [7] transformed target and source images into the $l\alpha\beta$ color space first. They used the Gaussian Mixture Models- Exception Maximum (GMM-EM) method to cluster the source color image into specified areas and use the K-means algorithm to segment the target image. They then imposed mean chromaticity values from the source image onto corresponding areas in the target image. C. C. Liu and G. N. Hu [11] considered the histogram in each plane of a color space as a probability density function. They first transformed the target and source images into LAB color space and then computed the probability density characteristics of each plane in each image to find the correlation function for every two corresponding planes between the target and source images. The transferred pixel values for each plane are determined by conducting the corresponding correlativity function on each target image plane. The color transferred target image is finally constructed by replacing each plane with its corresponding transferred plane. G. R. Greenfield and D. H. House [12] proposed an approach to recolor a target image according to the color scheme from the source image. The target and source images are first decomposed into clusters of pixels with similar color. The color palette for each image is constructed by selecting the most typical colors from the above decompositions. The color transfer from source image to the target image is evaluated by matching the decomposed areas between the target and source decompositions using a Euclidean metric. The resulting quality of the color transferred image is highly dependent on the previous segmentations. Zhou et al. [13] synthesized a natural scene using multiple labeled level eigen-spaces to generate a depth map, using multiple color level eigen-spaces to generate natural textures to fill in the label maps. Their approach can model natural textures and common depth layouts without learning large numbers of parameters to synthesize perceptually natural scene images. However, they often needed more than fifty sample images (depending on the complexity of the sample image) to train their system to extract sample textures and eigen-spaces. The goal of this paper is to find a simple and efficient global color transferring approach and a measurement metrics to evaluate color transferring scheme performance.

RBFN is a statistical method used to model the relationships between several independent variables and a dependent variable by fitting an equation to the observed data. Numerous experiments have shown that RBFN can be used to make accurate predictions. RBFN procedures are very widely used in the social and natural sciences today [14]. It is also a suitable technique for us in global color transferring from a source image to a target image. The RGB color space is the most popular space used in sensor and display devices. It is reasonable to take use multiple regression analysis on each component in a color image in the RGB domain. RBFN is conducted in this work on the RGB domains of two images for global color transferring. The main stages in the proposed algorithm are; (A) each plane's RBFN is trained by the data sampled from each plane of target and source images, (B) The color transfer is turned into a pixel value evaluation by conducting RBFN on the target image. The experimental results show that the proposed algorithm is effective and validated. The remainder of this paper is organized as follows: Section 2 presents the proposed algorithm. Section 3 describes the Empirical results. Section 4 concludes this paper.

2. The proposed algorithm

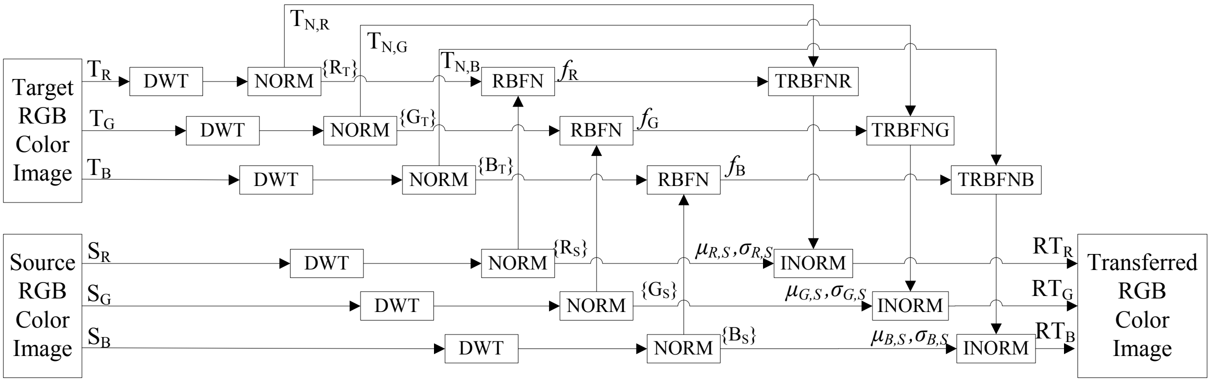
This paper presents a way to transfer the colors from a source RGB color image to a target RGB color image. Fig. 1 is flow chart of the recolor for a target RGB color image with multiple regression analysis. In order to obtain a more accurate color transfer result, the pixel intensities in each plane of both the target image and source image are normalized and sorted. The normalized and sorted data on each plane are then conducted to train the corresponding RBFN for each plane, respectively. These trained RBFN are used to determine the new values for each pixel in the target image on each plane. These new values are then combined into the RGB result color image. Detailed descriptions of the proposed scheme are illustrated as follows.

2.1 The RGB color space.

The color of an object can be explained scientifically as the color of light reflected by the particular object. Although the range of colors is infinite, a full range of perceivable colors by human eyes is able to be reproduced by combining different proportions of the three primarily colors; red, green, and blue [15].

In televisions, computer monitors, and colored image projection systems, by using only the three colors are enough to adequately represent any of the unlimited visible colors [16]. The image in RGB color space is the most suitable for the color representation of a color image. The RGB color model is the most popular and natural color model, because it can compose any color adequately. R, G and B component of a color in RGB color space are given by:

$$R,G,B = k\int_{400}^{700} I[\lambda]\Phi[\lambda]S_{R,G,B}[\lambda]d\lambda \qquad (1)$$



TRBFNR: Trained Radius Basis Function Network for R-component
TRBFNG: Trained Radius Basis Function Network for G-component
TRBFNB: Trained Radius Basis Function Network for B-component
RBFN: Radial basis function network
NORM: Normalization operation
INORM: Inverse normalization operation
DWT: Discrete Wavelet Transform

Fig. 1. The flow chart of the proposed color transfer algorithm.

where k is a constant that defines the total overall brightness response of the human eyes, $I[\lambda]$ is the illumination spectral intensity of a color, $\Phi[\lambda]$ is the object spectral reflectivity, $S_{R,G,B}[\lambda]$ is the spectral sensitivity of the R or G or B channel of the detector and λ is the wavelength. The normalized red, green, and blue coordinates are defined as follow:

$$r = R/(R+G+B) \tag{2}$$

$$g = G/(R+G+B) \tag{3}$$

$$b = B/(R+G+B) \tag{4}$$

where R, G and B are the intensities of red, green, and blue light at a given pixel [17].

2.2 Discrete Wavelet Transform

The discrete wavelet transform (DWT) is a fast, linear, invertible and orthogonal operation, just like the Discrete Fourier transform (DFT). The basic idea lying under the discrete wavelet transform is to define a time-scale representation of a signal (unlike short time fourier transform (STFT) which defines a time-frequency signal representation) by decomposing it onto a set of basis functions, called wavelet. Wavelets are obtained from a single prototype wavelet, called mother wavelet, by dilations and contractions, that is, scaling, and shifts [18,19]. DWT is suitable for the analysis of non-stationary signals since it allows simultaneous localization in time and in scale, unlike STFT which uses fixed time-frequency resolution and thus allows localization only in time or in frequency.

Application of DWT results in a multilevel decomposition of the input signal into high and low frequency components in different resolutions according to the number of levels employed, as shown in Figure 2.a Let $H(\omega)$ and $G(\omega)$:

$$H(\omega) = \sum_k h_k e^{-jk\omega} \tag{5}$$

$$G(\omega) = \sum_k g_k e^{-jk\omega} \tag{6}$$

be a lowpass and highpass filter, respectively satisfying the orthogonality condition:

$$|H(\omega)|^2 + |G(\omega)|^2 = 1 \tag{7}$$

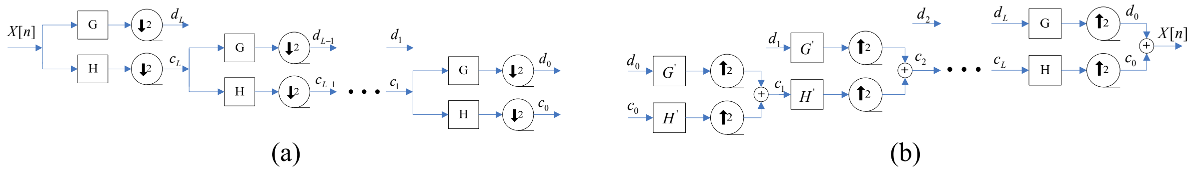
necessary for reconstruction capabilities of the transform. The filters $H(\omega)$ and $G(\omega)$ are also known as quadrature mirror filters. A signal $x[n]$ can be decomposed recursively according to [20, 21]:

$$c_{j-1,k} = \sum_n h_{n-2k} c_{j,n} \tag{8}$$

$$d_{j-1,k} = \sum_n g_{n-2k} c_{j,n} \tag{9}$$

which correspond to convolutions followed by downsampling by 2, as can be seen in fig 2 (a). h_k and g_k are the impulse responses of the lowpass and highpass filters, respectively. Index j spans the number if decomposition levels and lies in the range $[0, L+1]$, where $L+1$ represents the index of the high resolution level of the transform and 0 represents the index of the low resolution level. $c_{L+1,K}$ is equal to the input signal $x[k]$. The coefficients $c_{0,k}, d_{0,k}, d_{1,k}, \cdots, d_{L-1,k}, d_{L,k}$ are called the DWT wavelet coefficients of $x[n]$. $c_{0,k}$ is the lowest resolution component of $x[n]$ containing lowpass, 'smooth' information and $d_{j,k}$ are the detail coefficients of $x[n]$ at various bands of frequencies. The signal $x[n]$ can now be reconstructed from its DWT coefficients by considering the recursive formula:

$$c_{j,n} = \sum_n h_{n-2k} c_{j-1,k} + \sum_n g_{n-2k} d_{j-1,k} \tag{10}$$



(a) (b)

Fig. 2. (a)multi-level 1D wavelet decomposition; (b) Multi-level 1D wavelet reconstruction.

The reconstruction process is illustrated in Figure 2.b and defines the inverse discrete wavelet transform (IDWT). This time, upsampling precedes filtering at each level of the transform. It is obvious that, in the discrete case, DWT and IDWT can be implemented by two-channel tree-structured filter banks (Figure 2).

The discrete wavelet transform does not have a single set of basis functions. There are many families of wavelets, the most known of them being the Haar and the Daubechies wavelets [22].For example, the frequency responses $H(\omega)$ and $G(\omega)$ of the Haar wavelet filters are given by:

$$H(\omega)=\frac{1}{2}+\frac{1}{2}e^{-j\omega} \tag{11}$$

$$G(\omega)=\frac{1}{2}-\frac{1}{2}e^{-j\omega} \tag{12}$$

The two-dimensional discrete wavelet transform and its inverse are extensions of the one-dimensional transform. They are simply implemented by using one-dimensional DWTs and IDWTs along each dimension n and m separately:

$$DWT_{n\times m}[x[n,m]]=DWT_{n}[DWT_{m}[x[n,m]]] \tag{13}$$

In this way, separable two-dimensional filters are only considered. Each level is characterized by tree detail coefficient components representing the horizontal, vertical and diagonal edges of the input image. The lowest level consists of the low-resolution lowpass version of the image.

2.3 Normalization

In the RGB color space, colors are represented by varying values of R-component, G-component and B-component. The values of each of the R-component, G-component and B-component are represented with 8-bit unsigned integers (uint8) on a scale from 0 to 255. The RBFN should not work since it becomes impossible to avoid the overflow when the data are represented with uint8. In RBFN stage, the values of R-component, G-component and B-component are normalized to compress and limit the dynamic range to avoid the overflow. The mean and standard deviation of R-plane, G-plane and B-plane are evaluated, respectively. And then the values of R-component, G-component and B-component of a $M\times N$ RGB color image are normalized by the following equations.

$$\mu_X=\sum_{i=1}^{M}\sum_{j=1}^{N}X(i,j)/(M\times N), X\in\{R,G,B\} \tag{14}$$

$$\sigma_X=\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(X(i,j)-\mu_X)^2/(M\times N)}, X\in\{R,G,B\} \tag{15}$$

$$X_N(i,j)=(X(i,j)-\mu_X)/\sigma_X, X\in\{R,G,B\}, 1\le j\le M, 1\le j \tag{16}$$

where $X(i,j), X\in\{R,G,B\}$ is the pixel value of pixel $p(i,j)$ in the X plane of original RGB color image, and $X_N(i,j), X\in\{R,G,B\}$ is the corresponding pixel value of pixel $p(i,j)$ in the X plane of normalized RGB color image.

2.4 Radial basis function network (RBFN)

When performing experiments, we frequently obtain observations tabulated in the form of ordered pairs, $S=\{(\vec{x}_i,y_i)|i=1,2,\ldots n\}$, where $\vec{x}_i=\langle x_{i1},x_{i2},\ldots,x_{ip}\rangle^T$ is a column vector of dimension p and y_i is an scale. Given the observations, it is then usually desirable to be able to predict y from $\vec{x}$ by finding a mathematical model, that is, a function $y=f(\vec{x})$ that fits the observations as closely as possible. Finding a model of a function $y=f(\vec{x})$ from a set of known points (training set) is a function approximation.

Radial Basis Function Network (RBFN) is one of the most promising and utilized scheme for function approximations [14]. The RBFN is a special class of single hidden layer feed-forward neural network with three layers, namely the input layer, the hidden layer, and the output layer as shown in Fig. 3. The input layer distributes the components of the input vector to each of the neurons in the hidden layer. Each neuron in the hidden layer then generates the activation in form of the associated radial basis function. Each neuron in the output layer finally evaluates the weighted sum of the activations of the hidden neurons, and the general mathematical form of the output nodes is as follows:

$$y(\vec{x}_i) = f(\vec{x}_i) = \sum_{j=1}^{q} w_j \psi_j(\vec{x}_i), \qquad (17)$$

where $\psi_j(\vec{x}_i)$ are basis functions, and the most popular basis function is the Gaussian function with the expression::

$$\psi_j(\vec{x};\vec{\mu}_j,\sigma_j) = \exp\left(-\|\vec{x}-\vec{\mu}_j\|^2 / \sigma_j^2\right), \quad j = 1,2,\ldots,q, \qquad (18)$$

where $\vec{\mu}_j$ is the n -dimensional vector stands for the center of the $j-th$ radial basis function for the $j-th$ neuron in the hidden layer, σ_j is the width parameter of the $j-th$ radial basis function , and $\|\bullet\|$ is the Euclidean norm.

An RBFN outputs for a given input vector is completely determined by the number of activation functions, the activation functions of the hidden neurons $\psi_j(\vec{x};\vec{\mu}_j,\sigma_j)$, and the weights w_j that link the hidden layer neurons and the output. On the other hand, to train a RBFN is to obtain the values of three sets of parameters: the centers $\{\vec{\mu}_j\}$, the bandwidths $\{\sigma_j\}$, and the weights, so that to minimize the sum of absolute error between the true output and the output of the network for the corresponding input. Once the values of three sets of parameters are determined, and the number of neurons in the hidden layer is determined by the follows: the RBFN starts with one radial function, and increases the number of radial functions in the hidden layer one by one until no performance of the RBFN is improved by adding a new radial function.

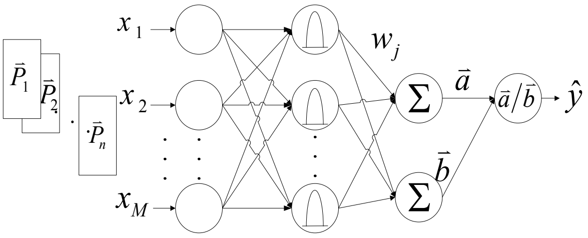
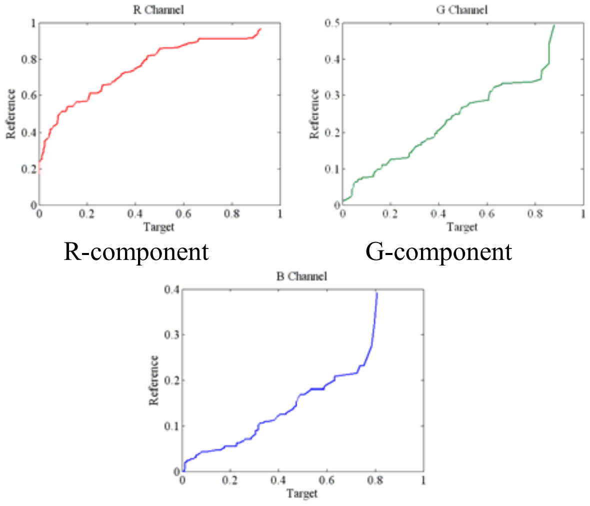


Fig. 3. Structure of RBFN.



R-component G-component

B-component

Fig. 4. The curves of intensity transfer function in each plane obtained by RBFN.

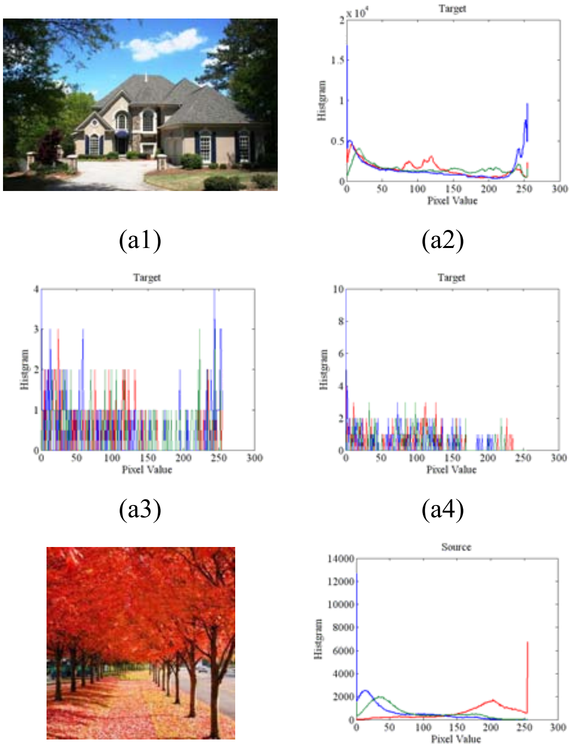
The training of RBFN is used to train the RBFNs of the three elements R, G, and B of these color-block images. These trained RBFNs are used to find out the transferred R, G, and B values of each pixel of the target image. These transferred values are constructed into the transferred target RGB color image. Fig. 4 shows the curves of intensity transferring functions of component R, G, and B between source image and target image.

3. Experiment results

These experiments were conducted on a computer with a 2.8GHz Intel Pentium processor and 2 GB RAM running Matlab version 7.6. For 800×504 target image and 380×391 source image, the color transferring procedure lasted about 0.325 seconds and the training procedure lasted from 0.54 to 1.36 seconds; the time consumptions in color transferring and RBFN training are significantly dependent on the number of radial basis functions. If the number is too low, the network may not calculate a proper estimation of the data. On the other hand, if too many hidden layer units are used, the network tends to overfit the training data. In addition, the number of radial basis functions is determined by the centers and the widths (or spreads) of the Gaussian density function in constructing RBFN, and the assigned error between the output of the network and the correct output.

This section presents experimental results under various conditions to illustrate the utility and efficiency of the proposed scheme. The input target RGB color image is an outdoor scene shot in a sunny winter day with size 800×504 pixels and the input source RGB color image is a dusk scene with size 380×391 pixels as shown in Fig.5.

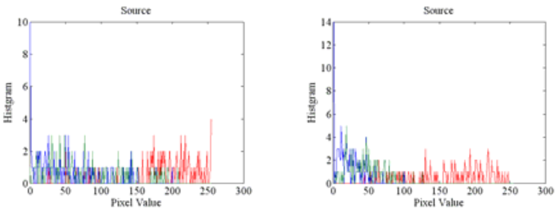
Due to the numerical differences between colors in the CIELAB system is very consistent with human visual perceptions; the color distance in terms of CIELAB components really indicates how much the color transferred image differs from the source image. To present the color distance between the transferred image and source image with CIELAB component units is the most suitable way to measure the performance of color transferring schemes. In this paper, the color distance measured with CIELAB component units was conducted to measure the color transferring performances.



(a1) (a2)

(a3) (a4)

(b1) (b2)



(b3) (b4)

Fig. 5. The target image and source image.

In order to give a detailed description of the color transfer results, several measurement metrics (MM) are conducted to measure the color transfer performance of the proposed algorithm. They are the difference in mean values of the color transferred image from the source image in lightness/ darkness ($L*$), in red-shade/ green-shade ($a*$), in yellow-shade/ blue-shade ($b*$), in chromaticity ($C*$), in hue ($H*$), and in total color ($E*$) [23]. These performance measures are based on two images: a source color image and the color transferred image. These performance measures are described as follows:

$$C* = \sqrt{(a*)^2 + (b*)^2} \tag{19}$$

$$E* = \sqrt{(L*)^2 + (a*)^2 + (b*)^2} \tag{20}$$

$$H* = (180 \times \arctan 2(b*/a*))/\pi \tag{21}$$

$$\overline{X} = \sum_{i=1}^{M}\sum_{j=1}^{N} X(i,j)/(M \times N), X \in \{L*, a*, b*, C*, H*, E*\} \tag{22}$$

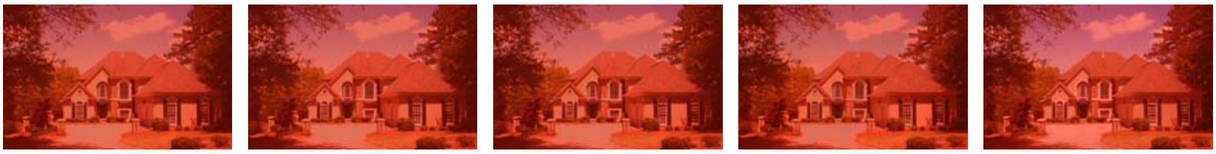
$$\Delta\overline{X} = \overline{X}_t - \overline{X}_s \tag{23}$$

$$\Delta\overline{X}(\%) = 100 \times (|\Delta\overline{X}|/\overline{X}_s) \tag{24}$$

where $X(i, j), X \in \{L*, a*, b*, C*, H*, E*\}$ is the pixel value of pixel $p(i, j)$ in the X plane of original CIELAB color image, $M \times N$ is the size of the test image, $\overline{X}$ is the mean of X, $\overline{X}_t$ is the mean of X component of color transferred image, $\overline{X}_s$ is the mean of X component of source image, $\Delta\overline{X}$ is the difference of X component mean of the transferred image from the source image, $\Delta\overline{X}(\%)$ is the difference ratio of X component mean of the transferred image from the source image, and arctan2 is a more novel version (four quadrant) of the arc-tangent function that returns the angle in the full range (− π, π], and is defined as the following equation [24]:

$$\arctan 2\left(\frac{y}{x}\right)=\begin{cases}\arctan(y/x), x>0\\ \pi+\arctan(y/x), x<0, y\geq 0\\ -\pi+\arctan(y/x), x<0, y<0\\ \pi/2, x=0, y>0\\ -\pi/2, x=0, y<0\\ \mathit{undefined}\ , x=y=0.\end{cases} \quad (25)$$

For testing images, these measurement metrics are evaluated by applying the proposed algorithm in the RGB domain. These measurement metrics are listed in tables for the performance analysis. Fig. 6 and Tables 1- 3 are used to demonstrate the color transfer results corresponding to the variation in the assigned error (AE) between the output of the network and the correct output with 100 centers and width 1.



(a) error=1 (b) error =0.1 (c)error =0.01 (d)error =0.001 (e) error =0.0002

Fig.6. Visible results with variation in the assigned error.

Table 1. Quantities vary with respective to the assigned error

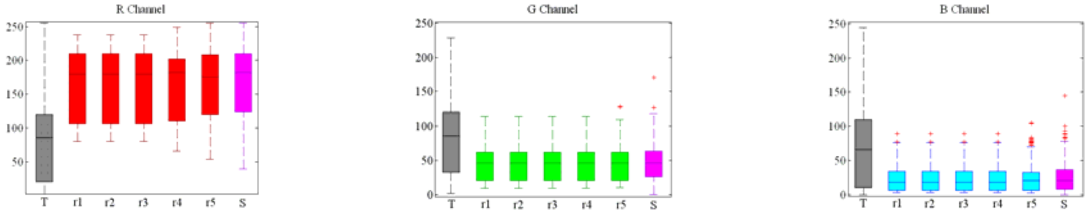
Assign error(AE)	Number of RBF			Mean-Square-Error			Training time(sec.)	Simulation time(sec.)
	R	G	B	R	G	B		
1	2	2	2	0.001429	0.000206	0.0002331	0.31	0.54
0.1	2	2	2	0.001429	0.000206	0.0002331	0.31	0.55
0.01	2	2	2	0.001429	0.000206	0.0002331	0.31	0.66
0.001	5	2	2	0.0007037	0.000206	0.0002331	0.34	0.89
0.0002	9	5	3	0.0001329	0.0001994	0.0001944	0.33	1.36

Table 2. Statistical values of color components of target image, source image and color transferred images with various AE.

Statistical Values	R		G		B		C		H		E	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
Target	94.07	71.94	110.38	77.07	111.56	93.25	177.37	15.31	63.36	4.15	220.98	38.85
Source	180.71	56.1	73.19	55.68	49.42	50.05	237.31	23.62	63.04	1.52	269.69	28.48
AE=1	169.74	52.79	56.12	33.96	38.99	34.21	239.46	12.88	62.05	1.00	263.4	23.22
AE=0.1	169.74	52.79	56.12	33.96	38.99	34.21	239.46	12.88	62.05	1.00	263.4	23.22
AE=0.01	169.74	52.79	56.12	33.96	38.99	34.21	239.46	12.88	62.05	1.00	263.4	23.22
AE=0.001	170.88	52.38	56.12	33.96	38.99	34.21	239.83	13.86	62.05	0.99	263.96	23.43
AE=0.0002	172.09	52.83	56.56	35.03	43.18	40.29	238.95	14.12	61.67	1.6	263.67	23.19

Table 3. Difference from transferred image to source image with respective to the assigned error

	ΔR(%)	ΔG(%)	ΔB(%)	ΔC(%)	ΔH(%)	ΔE(%)
AE=1	6.02	23.32	21.1	0.91	1.57	2.33
AE=0.1	6.02	23.32	21.1	0.91	1.57	2.33
AE=0.01	6.02	23.32	21.1	0.91	1.57	2.33
AE=0.001	5.44	23.32	21.1	1.06	1.57	2.12
AE=0.0002	4.77	22.72	12.63	0.69	2.17	2.32



R G B

Fig. 7. The box-plots of target image, source image and color transferred images.

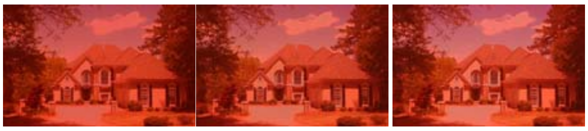
Fig. 7 shows the corresponding box-plot of the previous experiment. Fig.8 and Tables 4- 5 are used to demonstrate the color transfer results corresponding to the variation in the number of centers with width 1 and assigned error 0.001. Fig.9 and Tables 6- 7 are used to demonstrate the color transfer results corresponding to the variation in width with 100 centers and assigned error 0.001. We can make the following conclusions based on the above Figures and Tables: (A) The time consumption on training RBFN and on the simulation is significantly dependent on the number of RBFs, and the number of RBFs is determined by the assign error and the number of centers. (B) The statistical values (mean, standard deviation (STD)) of the measurement metrics for each image in L*, a*, b*, C*, H* and E* show that these statistical values are almost the same between the source images and the corresponding transferred images obtained using the proposed scheme. (C)The mean square error between the simulation and sources images is decreasing when the number of RBFNs is increasing. (D)The performance of larger width RBF is superior to the smaller width RBF

4. Conclusions

Transferring a source image's color to a target image involves changes the target image's color to enhance the target image's color features. In this paper, a radial basis function network (RBFN) based on discrete wavelet transform algorithm for global color transferring between images is proposed. RBFN is trained on the R, G, and B planes to obtain the corresponding trained RBFN for each plane, respectively. The transferred intensity of each pixel on each plane of the target image is then evaluated using the trained RBFN.

Table 4. Quantities vary with respective to the assigned error

Number of centers	Number of RBF			Mean-Square-Error			Training time(sec.)	Simulation time(sec.)
	R	G	B	R	G	B		
50	5	2	2	0.0009792	0.0003074	0.0001499	0.30	0.74
100	5	2	2	0.0007037	0.000206	0.0002331	0.34	0.89
200	6	2	2	0.0007474	0.0005095	0.0005309	0.31	0.77



Centers=50 Centers=100 Centers=200

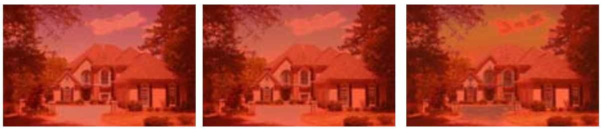
Fig.8. Visible results with variation in the number of centers.

Table 5. Difference from transferred image to source image with respective to the assigned error

Centers	ΔR(%)	ΔG(%)	ΔB(%)	ΔC(%)	ΔH(%)	ΔE(%)
50	7.35	32.07	32.15	1.64	1.71	2.37
100	5.44	23.32	21.1	1.06	1.57	2.12
200	3.1	19.46	12.08	0.98	1.89	1.61

Table 6. Quantities vary with respective to the width of RBFN

Width	Number of RBF			Mean-Square-Error			Training time(sec.)	Simulation time(sec.)
	R	G	B	R	G	B		
1	5	2	2	0.0007037	0.000206	0.0002331	0.34	0.89
0.5	7	2	2	0.000822	0.0002685	0.0002884	0.31	0.99
0.25	5	3	3	0.0007244	0.0008659	0.0006102	0.30	0.93



width=1 width=0.5 width=0.25

Fig.9. Visible results with variation in the width.

Table 7. Difference from transferred image to source image with respective to the width

Width	ΔR(%)	ΔG(%)	ΔB(%)	ΔC(%)	ΔH(%)	ΔE(%)
1	5.44	23.32	21.1	1.06	1.57	2.12
0.5	5.49	24.13	28.63	1.61	1.11	1.81
0.25	6.83	24.61	43.06	2.22	0.22	1.63

These transferred R, G, and B planes of the target image are then combined into the result color image result. The experimental results show that the proposed approach has three major advantages: (A) the proposed algorithm is manual free, simple, effective and accurate in transferring color between images without any change in the image details, (B) the proposed algorithm is time saving, due to operation directly and completely on RGB domain; it does not need the transformations among color spaces, (C) there are no restrictions in the image dynamic color ranges in the proposed algorithm, (D) the discrete wavelet transform can efficiently obtain feature pixels for training the RBFN. In the future, we will combine the proposed scheme with image segmentation schemes to improve the color transfer results.

References

[1] X. H. Han, Y. W. Chen and J. M. Lei, "A spatio-chromatic ICA based noise reduction in color images," *International Journal of Innovative Computing, Information and Control*, Vol. 4, No. 3, 2008 , pp. 661-670.

[2] Y. S. Lin, M. Z. Rau and S. J. Lee, "Applying self-constructing clustering to color image quantization," *ICIC Express Letters,* Vol. 3, No. 3(B), 2009, pp. 813-818.

[3] S. C. Huang, "A new scheme for generating initial palettes of color quantized images," *International Journal of Innovative Computing, Information and Control*, Vol. 6, No. 7, 2010, pp. 3011-3022.

[4] M. V. Sudhamani and C. R. Venugopal, "Nonparametric classification of data viz. clustering for extracting color features: An application for image retrieval," *ICIC Express Letters*, Vol. 1, No. 1, 2007, pp. 15-20.

[5] T. Pouli, E. Reinhard, "Progressive histogram reshaping for creative color transfer and tone reproduction," *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering 2010*, 2010, pp. 81- 90.

[6] S. De, A. Dagan, P. Roan, J. Rosen, M. Sinanan, M. Gupta, B. "Hannaford, CIELab and RGB color values of in vivo normal and grasped porcine liver," *Proceedings of Medicine Meets Virtual Reality (MMVR 15)* , 2007, pp. 109-111.

[7] S. Xu, S. Zhejiang, S. Zhang, and X. Ye, "Uniform color transfer," *Proceedings of IEEE International Conference on Image Processing*, 2005, pp. 940-943.

[8] D. Freedman and P. Kisilev, "Object-to-object color transfer: Optimal flows and SMSP transformations," *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2010, pp. 287 – 294.

[9] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications 21,* 2001, pp. 34–41.

[10] D.L. Ruderman, T.W. Cronin, and C.C. Chiao, "Statistics of cone responses to natural images: Implications for visual coding," *J. Optical Soc. of America,* Vol. 15, No. 8, 1998, pp. 2036-2045.

[11] C. C. Liu, G. N. Hu, "A re-coloring algorithm for a color image using statistic scheme in CIE L* a* b* color space," *2010 International Symposium on Computer, Communication, Control and Automation,* May 5-7, 2010, pp. 240-243.

[12] G. R. Greenfield, and D. H. House, "Image recoloring induced by palette color associations," *Journal of WSCG,* Vol. 11, No. 1, 2003, pp. 189-196.

[13] Q. Zhou, L. Ma, M. Celenk, D. "Chelberg, Natural scene synthesis using multiple eigenspaces," *2003 International Conference on Image Processing 3,* 2003, pp. II - 121-124.

[14] T. Ando, S. Konishi, S. Imoto, "Nonlinear regression modeling via regularized radial basis function networks," *Journal of Statistical*

Planning and Inference, Vol.138, 2008, pp. 3616- 3633.

[15] S.S. Deeb, "Genetics of variation in human color vision and the retinal cone mosaic," *Current Opinion in Genetics & Development,* Vol. 16, No. 3, 2006, pp. 301– 307.

[16] C. Clausen, H. Wechsler, "Color image compression using PCA and back propagation learning," *Pattern Recognition*, Vol. 33, 2000, pp. 1555–1560.

[17] I. Andreadis, Ph. Tsalides, "Analog computation of image chromaticity," *Real-Time Imaging*, Vol. 3, 1997, pp. 1–6.

[18] Caroline Chaux, "Image Analysis Using a Dual-Tree M-Band Wavelet Transform," *IEEE Transactions on Image Processing* Aug. 2006, pp. 2397 – 2412.

[19] Ivan W. Selesnick, "A Higher Density Discrete Wavelet Transform," *IEEE Transactions on see also Acoustics, Speech, and Signal Processing, IEEE Transactions on Signal Processing.* Aug. 2006 , pp. 3039 – 3048.

[20] Sunil Lee, "Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform," *IEEE Transactions on Information Forensics and Security* , 2007, pp. 321-330.

[21] Pedro Henrique Cox and Aparecido Augusto de Carvalho, "Discrete Wavelet Transform Signal Analyzer," *IEEE Transactions on Instrumentation and Measurement* , 2007, pp. 1640 – 1647.

[22] Christian Tenllado, Member, Javier Setoain, Manuel Prieto, Luis Pin˜ uel, and Francisco Tirado, *Parallel Implementation of the 2D Discrete Wavelet Transform on Graphics Processing Units: Filter Bank versus Lifting*, Transactions on Parallel and Distributed Systems March , 2008, pp. 299 – 310.

[23] M. E. Celebi, H.A. Kingravi, F. Celiker, "Fast colour space transformations using minimax approximations," *IET Image Process.*, Vol. 4, No. 2, 2010, pp. 70–80.

[24] P. Djeu, M. Quinlan, and P. Stone, "Improving particle filter performance using SSE instructions," *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2009, pp. 3480- 3485.