ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

Exploring the Implementation of User Controlled Color Transfer Algorithm for Images

Aarti K. Masal, R.R.Dube

Abstract*— Color transfer algorithms aim to apply a color palette, mood or style from one image to another, operating either in a three dimensional color space, or splitting the problem into three one-dimensional problems. In this paper we present a interactive method for recoloring a destination image according to the color scheme found in a source image. Our recoloring method attempts to maintain the destination image's original value structure. This is accomplished by transferring RGB color space to lαβ color space. This paper presents a simple & interactive algorithm which allows user to select the objects on source & target image for recoloring. The proposed algorithm uses color statistics of source & target image. Target image's color influence map is prepared. It is a mask that specifies what parts of target image will be affected according to selected color range. After that the target image is recolored according to the color influence map. Our paper explores the software implementation of the proposed technique. The proposed algorithm is implemented in JAVA object oriented language.

***Index Terms*— Color Transfer, Local Color Statistics, Color Characteristics, Orthogonal Color Space, Color Influence Map.**

I. INTRODUCTION

Color transfer is a kind of process of color alteration, which changes an image's color to accord with another image's color characteristics. Applications of color transfer range from subtle post processing on images to improve their appearance to more dramatic alterations, such as converting a daylight image into a night scene. Reinhard and his colleagues presented a pioneering work about color transfer [Reinhard et al. 2001]. Because of the correlations between the three channels' values in RGB color space, they transform pixels' values from RGB to *l*ab color space, and then manipulate them separately, and finally return to RGB space. Their method can produce amazing results. This technique transfers color characteristics from a source to a target image. In the target image *It*, the transferred color at pixel *Ct* in the *lαβ* color space is:

$$g(C_t) = \mu_s + \frac{\sigma_s}{\sigma_t}(C_t - \mu_t)$$

Where μs, μt are the means of the underlying Gaussian distribution in the lαβ color space of the respective source and target images. σs, σt are the respective standard deviations. We call this approach global color transfer because the color statistics are calculated by taking into account all pixels in the respective images. Typically, there are two types of color transfer method: global color transfer and local color transfer. Most of existing methods are aimed to perform global color transfer. This usually means that the whole image is affected. Global color transfer does not adequate spatial consideration, so it cannot avoid the following two problems. One is that if the source or target image contains different color regions, the global transfer cannot distinguish the different statistics and will mix regions up. The other problem is that if the colors of the two images are very different, in the *lαβ* color space, the chromaticity channels are easily exaggerated which will cause unnatural and saturated result. Later to overcome problems associate with global color transfer local color transfer methods have been proposed for application to specific local regions within an image. In this case, the user can select a pair of corresponding regions in the source and target image for the color change.

(a) Source image (b) Target image (c) Resulting image

Fig. 1: The effect of total image recoloring using the method of Reinhard et al [1].

(a)Source image (b) Target image (c)Resulting image

Fig. 2: Partial image recoloring with an image fragment using proposed method.

Figure 1 shows the effect of global color transfer in which the whole image gets affected while figure 2 shows the partial image recoloring which is nothing but a local color transfer. Here the user just has to select a pair of corresponding regions in the source and target image for the color change. The data of the selected region is used to calculate its color statistics. We use them to estimate pixels of the target image belong, or rather are close enough, to

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

the selected color range. Then color transformation is applied to the image, according to this estimation. In this paper a novel approach towards the software implementation of the local color transfer algorithm is proposed.

II. Related work

Most of the existing color transfer algorithm performs global color transfer. Reinhard et al presented a method for global color transfer. [1] Shifts and scales the pixel values of the source image to match the mean and standard deviation from the target. This is done in the lab opponent color space, which is on averaged correlated [2]. This allows the transfer to take place independently in each channel, turning a potentially complex 3D problem into three much simpler 1D problems. Although this technique can be successful for a large range of images, the quality of the results largely depend on the composition of the source and target images. While this algorithm is simple and efficient, it cannot distinguish the color statistics of local regions, which generate an unnatural or over saturation in resulting image. Some authors try to solve this problem using complex image spatial or color characteristics, but these methods have other limitations. Method based on Basic Color Categories [3] is limited in variations of color changing, because any color can be replaced only by a color from the same color category. For example, we can't turn blue into red. Another method, described in [4], uses complex image color and spatial characteristics to determine image palette associations. Image color segmentation using Expectation Maximization method is offered in [6] to solve the problem of local color transfer, but segmentation and region color decision is performed fully automatically, and again for the whole image, which is not always desirable. Cellular automata is used successfully in [7] to select an object of interest, but only a single color can be used as color source and recoloring is very uniform, even when some variability of color shade is desired according to initial look and feel of the object.

III. Local color transfer

This method is developed to correct an object on a *target* image, selected by user. It is important, that user doesn't have to select the object by shape, for it is difficult to do precisely. The object is selected in terms of its color range and calculating its color statistics. The source color can be defined as a single color or a region of another image, that we will call the *source* image. In this case source color statistics are calculated. After the information of the target color range is gathered, we prepare the target image's Color Influence Map (CIM). It is a mask that specifies what parts of the target image will be affected according to the selected color range.

The next step is recoloring itself that is applied to the target image according to the prepared CIM. To recolor the target image we use a modified version of the Color Transfer algorithm, described in [1]. The basic algorithm uses transformation method, based on source and target color statistics in lαβ color space [1],[4]. The limitations of this method are 1) only an image can be used as a source for recoloring; 2) only the whole image can be recolored. These limitations are removed in our algorithm.

A. ***Need to transfer of RGB color space to lαβ color space:*** When a typical three channel image is represented in any of the most well-known color spaces, there will be correlations between the different channels' values. For example, in RGB space, most pixels will have large values for the red and green channel if the blue channel is large. This implies that if we want to change the appearance of a pixel's color in a coherent way, we must modify all color channels in tandem. This complicates any color modification process. What we want is an orthogonal color space without correlations between the axes. When a typical three channel image is represented in any of the most well-known color spaces, there will be correlations between the different channels' values. For example, in RGB space, most pixels will have large values for the red and green channel if the blue channel is large. This implies that if we want to change the appearance of a pixel's color in a coherent way, we must modify all color channels in tandem. This complicates any color modification process. What we want is an orthogonal color space without correlations between the axes which lets us apply different operations in different color channels with some confidence that undesirable cross channel artifacts won't occur. Additionally, this color space is logarithmic, which means to a first approximation that uniform changes in channel intensity tend to be equally detectable. Once the RGB color space is transformed to to lαβ color space then the object's color statistics is calculated.

B. ***Transformation from RGB color space to lαβ color space*:** Our goal is to manipulate RGB images, which are often of unknown phosphor chromaticity; we first show a reasonable method of converting RGB signals to Ruder man et al.'s perception-based color space lαβ. Because lαβ is a transform of LMS cone space, we first convert the image to LMS space in two steps. The first is a conversion from RGB to XYZ tri stimulus values. This conversion depends on the phosphors of the monitor that the image was originally intended to be displayed on. Because that information is rarely available, we use a device-independent conversion that maps white in the chromaticity diagram (CIE xy) to white in RGB space and vice versa. Because we define white in the chromaticity diagram as $x = X/(X + Y + Z) = 0.333$, $y = Y/(X + Y + Z) = 0.333$, we need a transform that maps $X = Y = Z = 1$ to $R = G = B = 1$. To achieve this,

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

we modified the XYZitu601-1 (D65) standard conversion matrix to have rows that add up to 1. The International Telecommunications Union standard matrix is (1)

$$M_{itu} = \begin{bmatrix} 0.4306 & 0.3415 & 0.1784 \\ 0.2220 & 0.7067 & 0.0713 \\ 0.0202 & 0.1295 & 0.9394 \end{bmatrix} \quad (1)$$

By letting *Mitux* = (111)*T* and solving for *x*, we obtain a vector *x* that we can use to multiply the columns of matrix *Mitu*, yielding the desired RGB to XYZ conversion:(2)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.5141 & 0.3239 & 0.1604 \\ 0.2651 & 0.6702 & 0.0641 \\ 0.0241 & 0.1228 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

Compared with *Mitu*, this normalization procedure constitutes a small adjustment to the matrix's values. Once in device-independent XYZ space, we can convert the image to LMS space using the following conversion:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3897 & 0.6890 & -0.0787 \\ -0.2298 & 1.1834 & 0.0464 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3)$$

Combining these two matrices gives the following transformation between RGB and LMS cone space:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

The data in this color space shows a great deal of skew, which we can largely eliminate by converting the data to logarithmic space:

$L = \log L$
$M = \log M$
$S = \log S$ (5)

Using an ensemble of spectral images that represents a good cross-section of naturally occurring images, Ruder man et al. proceed to decorrelate these axes. Their motivation was to better understand the human visual system, which they assumed would attempt to process input signals similarly. We can compute maximal decorrelation between the three axes using principal components analysis (PCA), which effectively rotates them. The three resulting orthogonal principal axes have simple forms and are close to having integer coefficients. Moving to those nearby integer coefficients, Ruderman et al. suggest the following transform:

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (6)$$

If we think of the *L* channel as red, the *M*as green, and the *S* as blue, we can see that this is a variant of many opponent-color models:

Achromatic $\alpha\ r + g + b$
Yellow–blue $\alpha\ r + g - b$
Red–green$\alpha r - g$ (7)

Thus the *l* axis represents an achromatic channel, while the α and β channels are chromatic yellow–blue and red–green opponent channels. The data in this space are symmetrical and compact, while we achieve decorrelation to higher than second order for the set of natural images tested.2 Flanagan et al.5 mentioned this color space earlier because, in this color space, the achromatic axis is orthogonal to the equiluminant plane. Our color-correction method operates in this *lαβ* space because decorrelation lets us treat the three color channels separately, simplifying the method.

After color processing, which we explain in the next section, we must transfer the result back to RGB to display it. For convenience, here are the inverse operations. We convert from *l*αβ to LMS using this matrix multiplication:

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} \quad (8)$$

Then, after raising the pixel values to the power ten to go back to linear space, we can convert the data from LMS to RGB using

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (9)$$

C. ***Calculating object's color statistics:*** At first user has to select an object to correct at the target image. This can be done by loosely selecting a rectangular region of the object that needs correction. There's no need to select the region precisely close to the edges of the area, that user wants to correct, because we will use its color range without spatial characteristics. The only limitation is that the whole region must be inside this object.

Fig. (3): The region is selected to correct the sky.

After the user has selected the region at the target image, we calculate color statistics (mean and variation) for this region, for each channel of the working color space separately:

$$\mu_c^R = \frac{1}{N_R} \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} c(i,j) \cdot \quad (10)$$

$$\sigma_c^R = \sqrt{\frac{1}{N_R - 1} \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \left(c(i,j) - E^R c\right)^2} \cdot \quad (11)$$

Where NR=(i_2-i_1) is the number of pixels in the selected region *R*, $1 \le i1 < i2 \le H$ and $1 \le j1 < j2 \le W$ define the rectangular region *R* and $c(i, j)$ is a processed color channel

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

of pixel (*i,j*).If colors of another image must be used as color source, then this step is applied to the source image as well. Note that all these and further calculations can be made in any color space, but as it is discussed earlier in this paper & is shown in [4] that the most suitable color space for color transformations is Rudermann et al. lαβ.

D. *Building the Color Influence Map*: After color statistics of the target image are gathered, they are used to determine the mask for the target image recoloring. CIM contains weights for color transformation for each pixel of the target image. Each pixel's weight is determined from its proximity to the color range, selected by user and stored in color statistics information. That is Mahalanobis distance between the pixel and the center of the color distribution, defined by the stored color statistics. Since lαβ is a color space with decorrelated axes (as stated in [1]) Mahalanobis distance turns to Euclidian:

$$\rho(x, \mu) = \| x - \mu^R \|_E \qquad (12)$$

where x – is a color vector in working color space and $\mu^R = (\mu^R l, \mu^R \alpha, \mu^R \beta)$, obtained from (10).

Weights *fij* in CIM are build using this formula:

$$fij = F(\rho(\mu^R, C(i,j))) \qquad (13)$$

where $C(i,j)$ is a color vector, $\rho(\mu^R, C(i,j))$ is obtained from (12), $F(x)$ is defined at $[0, \infty)$ and $\lim_{x \to \infty} F(x) = 0; F(0) = 1;$

ij

$F(x)$ is not strictly defined in (13), because there can be used various functions. Our experiments have shown that for photographs of good enough quality better results can be achieved using:

$$F(x) = e^{3x2} \qquad (14)$$

Fig. (4): CIM for the selected region from Fig. 2 using (14).

E. Color transfer: At this step the target image is recolored according to the prepared CIM. The basic color transformation uses the stored color statistics. The below original formula from [1] is applied to all the pixels of the target image, separately for each color channel. Here and further the source data are marked with index s and the target data are marked with index t.

$$c_t^{new}(i,j) = \mu_s + \frac{\sigma_s}{\sigma_t}(c_t(i,j) - \mu_t) \qquad (15)$$

This transformation is modified to include the CIM information and to apply a single color as a source, if necessary. Thus, if the source color is taken from an image, then the stored source color statistics are used:

$$c_t^{new}(i,j) = c_t(i,j) + f_{ij} \cdot \left(\mu_c^{R_s} + \frac{\sigma_c^{R_s}}{\sigma_c^{R_t}} \left(c_t(i,j) - \mu_c^{R_t} \right) - c_t(i,j) \right) \qquad (16)$$

$$i = \overline{1, H_t}, j = \overline{1, W_t},$$

Where $C_t^{new}(i,j)$, $C_t(i,j)$ are new and old values of a color channel of pixel (i,j) of the target image respectively; μ_c^{Rs}, μ_c^{Rt} are taken from (10); σ_c^{Rs}, σ_c^{Rt} are taken from (11), *fij* is taken from(13), *Ht* is target image height, *Wt* is target image width.

If the single color *Col* is used as the source for recoloring, then (16) turns to:

$$c_t^{new}(i,j) = c_t(i,j) + f_{ij} \cdot \left(Col_c - \mu_c^{R_t} \right), \qquad (17)$$

$$i = \overline{1, H_t}, j = \overline{1, W_t},$$

Fig. 2 shows the result for the image and selection. We can see that only the sky was corrected, while the field and the trees were left unchanged. Moreover, the details of the recolored area are saved, so that it has quite natural look. Fig. 5 shows the example of using a single color as source color. Like in the previous example, only the sky was corrected while the field and trees weren't damaged. And both Figs.2 and 5 shows, that the whole area of the selected color range was affected by color transformation. Note that in the lαβ color space luminance and chrominance information is separate, so it allows making image recoloring optional in this way easily.

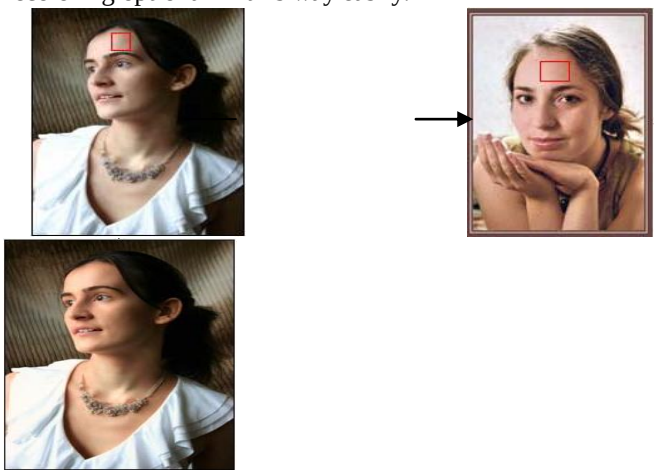

Fig.5: Skin color correction at a portrait. The model looks tanned at the result image.

IV. EXPERIMENTS

Here we try to implement the proposed algorithm in JAVA. In the first step of implementation a GUI is prepared which allows the user to load the source & target image of their interest. After that our algorithm calculates the histogram for both the source & target image and also it compares the histogram of source & target image. The source & target images are present in RGB color space. To perform the local color transfer it is required to convert the RGB color space into lαβ color space. The next step of implementation will be the conversion of RGB color space into lαβ color space. The source & target images which are

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

shown in fig 1 are loaded and after that our algorithm calculates the red, green, blue histogram for both the images. After calculation of histograms for both images comparison is performed. Figure (6) shows GUI through which user can load source & target images of their interest. It also shows the comparison of red histogram similarly our algorithm performs comparison for green as well as blue histogram. The next step is to convert RGB color space to lαβ color space. In this step the image is loaded and the algorithm performs the conversion from RGB color space to lαβ color space. The changes in the image are not perceptible to human eyes. Figure(7) shows the GUI for loading the image & then algorithm transfers RGB color space into lαβ color space.

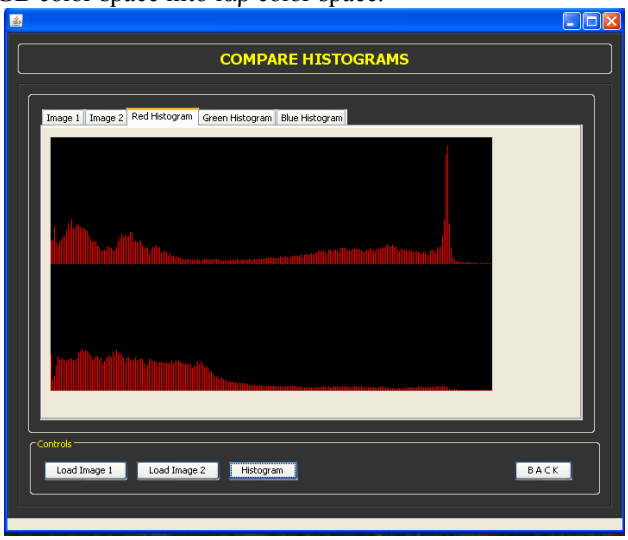

Fig. (6): GUI for red Histogram comparison

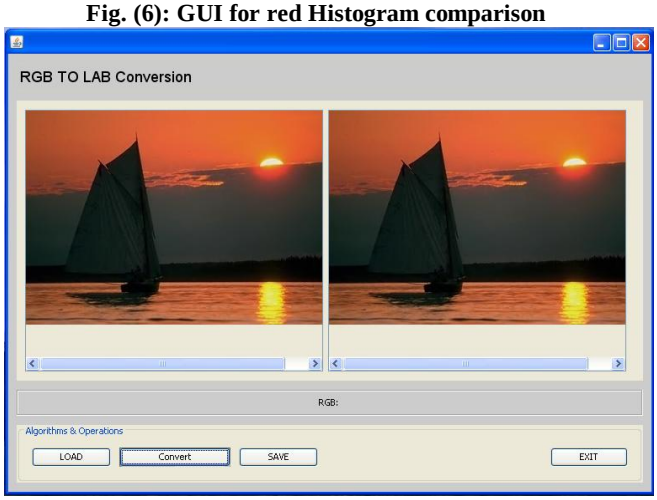

Fig. (7): GUI for RGB to lαβ color space transfer

V. CONCLUSION

In this paper, we discuss a color transfer method that can borrow one image's color characteristics from another also a novel approach for the implementation of interactive local color transfer algorithm is proposed. Here the color transfer is performed by mapping dominant colors of source image to target image. This algorithm allows user to correct an object of interest at an image saving one from the trouble of selecting it precisely. The proposed method allows the user to recolor a part of the image in a simple and intuitive way, preserving other color intact and achieving natural look of the result for wide variety of input images. The proposed algorithm is fast, fully automatic and can depict all the dominant color styles of the target image in the output image.

REFERENCES

[1] Eric Reinhard, Michael Ashikhmin, Bruce Gooch, Peter Shirley, "Color Transfer between Images", Computer Graphics and Applications, 2001, Vol.21, Issue 5, pp. 34-41.

[2] Ruderman D, Cronin T, Chiao C. Statistics of cone responses to n natural images: implications for visual coding. Journal of the Optical Society of America A 1998; 15(8):2036–45 .

[3] Youngha Chang, Suguru Saito, Masayuki Nakajima, "Color transformation based on Basic Color Categories of a Painting", Proceedings of Computer Graphics International, 2003, pp. 176-181.

[4] Donald H. House, Gary R. Greenfield, "Image Recoloring Induced by Palette Color Associations", Journal of WSCG, 2003, Vol.11, No.1, pp. 189-196.

[5] Daniel L. Rudermann, Thomas W. Cronin, Chuan-Chin Chiao, "Statistics of cone responses to natural images: implications for visual coding", Journal of the Optical Society of America, 1998, Vol.15, Issue 8, pp. 2036-2045.

[6] Jiaya Jia, Chi-Keung Tang, Yu-Wing Tai, "Local Color Transfer via Probabilistic Segmentation by Expectation-Maximization", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, Vol.1, pp. 747-754.

[7] V. Konushin, V. Vezhnevets, "Interactive Image Colorization and Recoloring Based on Coupled Map Lattices", Graphicon'2006 Conference Proceedings, 2006, pp.231-234.

[8] Alla Maslennikova, Vladimir Vezhnevets" Interactive Color Transfer between Images" Graphics and Media Lab., Department of Computational Mathematics and Cybernetics Lomonosov Moscow State University, Moscow, Russia.

AUTHOR BIOGRAPHY

Miss. Masal Aarti K. Student, ME (ELN), Walchand Institute of Technology, Solapur, Maharashtra, India.

Mr. R.R.Dube Assistance Professor at Walchand Institute of Technology, Solapur, Maharashtra, India