

An Approach towards Software Implementation of Interactive Color Transformation Algorithm

Miss. Aarti K. Masal¹, Asst. Prof. R.R.Dube²

Dept. of Electronics Engineering
Walchand Institute Of Technology, Solapur University, Solapur (Maharashtra), India

Abstract- In this paper we present a recoloring technique called interactive color transfer which is similar to local color transfer which gives a user more control to define local areas & match the local areas in order to perform color transfer. This paper presents a fast and simple interactive algorithm based on local color statistical analysis which is used to impose one image's color characteristics on another. Here focus on orthogonal color space i.e. $l\alpha\beta$ color space without correlation between the axes is given. Target image's color influence mask is prepared. It is a mask that specifies what parts of target image will be affected according to selected color range. After that target image is recolored in $l\alpha\beta$ color space according to prepared color influence map. In the $l\alpha\beta$ color space luminance and chrominance information is separate so it allows making image recoloring optional. The basic color transformation uses stored color statistics of source and target image. Our paper presents an approach towards the software implementation of the proposed technique. The proposed algorithm is implemented in JAVA object oriented language.

Index Terms- color space, recoloring, color transfer, local color information, image segmentation.

I. INTRODUCTION

Images can convey information not only through the depicted objects but also through the particular mood, color scheme and composition of the scene. Artists can manipulate the color palette manually to change the appearance of an image and achieve specific effects but that can be a time-consuming process, requiring advanced image manipulation skills. To that end, several color transfer techniques have been proposed that uses the color palette of a second image as a target and achieve similar results with minimal user input and skill necessary. Often this means removing a dominant and undesirable color cast, such as the yellow in photos taken under incandescent illumination. 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 have 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 color of the two images are very different, in the $l\alpha\beta$ 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.



Figure 1: (a) The effect of total image recoloring using the method of Reinhard et al [1].



Figure 1 : (b) Partial image recoloring with an image fragment using proposed method.

Figure 1(a) shows the effect of global color transfer in which the whole image gets affected while figure 1(b) shows the partial image recoloring which is nothing but a local color transfer. Here the user just have 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 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 perform 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 apotentially 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 oversaturation 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 $l\alpha\beta$ 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 $l\alpha\beta$ 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 $l\alpha\beta$ color space then the object's color statistics is calculated.

B. 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.



Figure (2): 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:

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

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

where $N_R=(i_2-i_1)$ is the number of pixels in the selected region R , $l \leq i1 < i2 \leq H$ and $l \leq j1 < j2 \leq W$ define the rectangular region R and $c(i, j)$ is a processed color channel 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. $l\alpha\beta$.

C. 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 $l\alpha\beta$ is a color space with decorrelated axes (as stated in [1]) Mahalanobis distance turns to Euclidian:

$$\rho(x, \mu) = \|x - \mu\|_E \quad (3)$$

where x – is a color vector in working color space and $\mu^R = (\mu^R_l, \mu^R_a, \mu^R_\beta)$, obtained from (1).

Weights f_{ij} in CIM are build using this formula:
 $f_{ij} = F(\rho(\mu^R, C(i,j)))$ (4)

where $C(i,j)$ is a color vector, $\rho(\mu^R, C(i,j))$ is obtained from (3), $F(x)$ is defined at $[0, \infty)$ and $\lim_{x \rightarrow \infty} F(x) = 0; F(0) = 1;$

$F(x)$ is not strictly defined in (4), because there can be used various functions. Our experiments have shown that for photographs of good enough quality better results can be achieved using:
 $F(x) = e^{-3x^2}$ (5)



Figure (3): CIM for the selected region from Fig. 2 using (5).

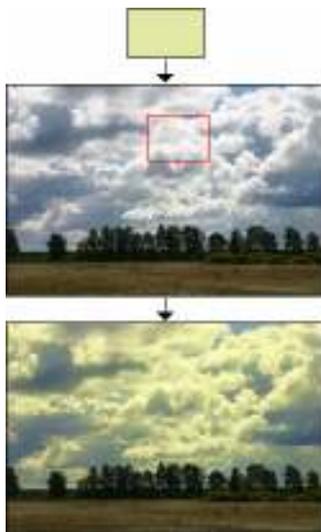


Figure (4): Partial image recoloring with a single color used as color source.

D. 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 .

$$c_t^{new}(i,j) = \mu_s + \frac{\sigma_t}{\sigma_s} (c_t(i,j) - \mu_t) \quad (6)$$

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:

$$c_t^{new}(i,j) = c_t(i,j) + f_{ij} \cdot \left(\mu_c^R + \frac{\sigma_c^R}{\sigma_c} (c_t(i,j) - \mu_c^R) - c_t(i,j) \right) \quad (7)$$

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

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^R, μ_c^{Rt} are taken from (1); $\sigma_c^R, \sigma_c^{Rt}$ are taken from (2), f_{ij} is taken from (4), H_t is target image height, W_t is target image width.

If the single color Col is used as the source for recoloring, then (7) turns to:

$$c_t^{new}(i,j) = c_t(i,j) + f_{ij} \cdot (Col - \mu_c^R) \quad (8)$$

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

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 $\lambda\beta$ color space luminance and chrominance information is separate, so it allows making image recoloring optional in this way easily.

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 $\lambda\beta$ color space. The next step of implementation will be the conversion of RGB color space into $\lambda\beta$ color space. The source & target images which are shown in fig 1(a) 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.

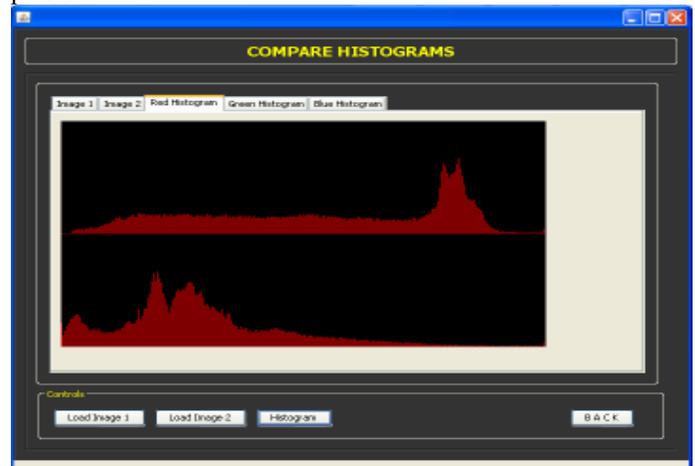


Figure (5): GUI for Histogram comparison

Figure(5) 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.

V. CONCLUSION

In this paper 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 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

First Author – Miss. Masal Aarti K., Pursuing ME(ELN), Walchand Institute of Technology, Solapur, aarti.masal@gmail.com.

Second Author – Mr. R.R.Dube, Assistant Professor(ME ELN), Walchand Institute of Technology, Solapur , rajendradube67@rediffmail.com .