International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.6, Nov-Dec. 2012 pp-4639-4642 ISSN: 2249-6645

User Interactive Color Transformation between Images

Miss. Aarti K. Masal, [1] Asst. Prof. R.R.Dube[2]

1,2Dept. of Electronics Engineering, Walchand Inistitsute of Technology, Sholapur University, Solapur (Maharashtra).

Abstract**:** *In this paper we present a process called color transfer which can borrow one image's color characteristics from another. Most current colorization algorithms either require a significant user effort or have large computational time. Here focus on orthogonal color space i.e. lαβ color space without correlation between the axes is given. Here we have implemented two global color transfer algorithms in lαβ color space using simple color statistical information such as mean, standard deviation and covariance between the pixels of image. Our approach is the extension of Reinhard's. Our local color transfer algorithm uses simple color statistical analysis to recolor the target image according to selected color range in source image. 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 lαβ color space according to prepared color influence map. In the lαβ 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. All the algorithms are implemented in JAVA object oriented language. The main advantage of proposed method over the existing one is it allows the user to recolor a part of the image in a simple & intuitive way, preserving other color intact & achieving natural look.

Index Terms: *color transfer, local color statistics, color characteristics, orthogonal color space, color influence map.*

I. INTRODUsCTION

Color is one of the main image attributes used in many different areas, such as medical image analysis, video object extraction, image compression, tracking system, art, photography and visualization for relaying information, or for conveying a specific mood. Nowadays, digital cameras are very popular, and most people take many photos on their trips, reunions, etc. However, since most end users are not professional photographers, a user usually takes a lot of photos but eventually finds out that only a small portion of them are satisfactory. Indeed, with powerful commercial post-processing software, the experts could enhance these defect photos, but this task can be time-consuming. The direct enhancement of the photos by using some post processing tools may not be an easy task. For example, in the case of a backlighted photo, one may think that the user can use the brightness/hue adjustment tools as contained in ordinary image processing software. However, such tools usually can only be used to adjust the brightness/hue globally. If such tools are applied to achieve a brighter foreground, the background might be overexposed. Even if the foreground region in interest is carefully specified, the process is not only time-consuming but also may result in artifacts at the foreground and background border. On the other hand, if we can refer to the same object in other good quality photos, it would be much easier. Considering such users, color transfer forms a class of techniques that allow the color palette of an image to be altered using a second image as a reference. The task is then to select a reference image whose colors are preferred. Subsequently the algorithm will modify the original image such that it acquires the palette of that reference. In essence this operation can be seen as a function that, given two images,

produces a third that has maintained the semantic content of the one while acquiring the colors of the second. There are two types of color transfer between images, global color transfer and local color transfer. Most of the available algorithms either perform a global color transfer or local color transfer. In our work we have implemented both types of color transfer which are user interactive.

Figure 1: (a) Target Image, (b) Source image, (c) resulting image using proposed first global color transfer algorithm , (d)resulting image using proposed second global color transfer algorithm.

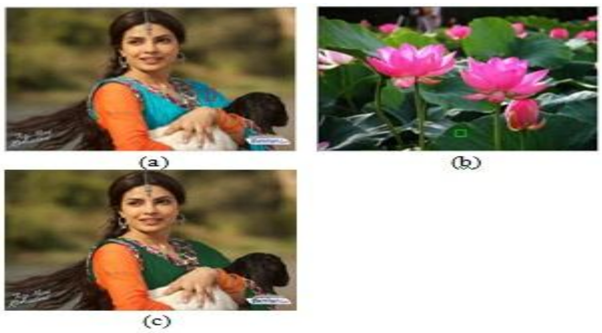

Figure 2: (a) Target Image, (b) Source image, (c) resulting image of local color transfer using proposed algorithm

.

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 the selected color range. Then color transformation is applied to the image, according to this estimation. In this paper we present color transfer algorithms which are user interactive and perform both types of color transfer global as well as local color transfer.

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 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.

Maslennikova designed a weighted mask of influence map for each pixel of the target image using a color influence map (CIM)[8] that specifies which part of the target image will be affected according to the selected color. Singular value decomposition algorithm used in[9] to decompose the covariance matrix which represents the covariance between source image’s pixel and also between the pixels of target image. It gives a method for global color transfer but it only works with the images of similar composition.

III. Global color transfer (first algorithm)

Color transfer is carried out by a statistics-based method that can achieve the goal in lαβ color space just utilizing the statistics -mean and covariance matrix. Here the covariance matrix can be deemed to the extension of standard deviation in decorrelated space. Data points of source image is scaled, rotated and translated to fit data points' cluster of target image in lαβ color space.

First, calculate the mean of pixel data along the three axes and the covariance matrix between the three components in color space for both the source and target images. All of the statistics are denoted by (lsrcmean; αsrcmean; βsrcmean), (ltgtmean; αtgtmean; βtgtmean), Covsrc and Covtgt , respectively.

Then decomposition of covariance matrices is carried out by Singular value decomposition algorithm. Singular value decomposition algorithm is factorization of a real or complex matrix [9].

$$Cov = U.\wedge.V^{T} \qquad (1)$$

Where U and V are orthogonal matrices and are composed of the eigenvectors of Cov, $\wedge = diag(\lambda^{l}; \lambda^{\alpha}; \lambda^{\beta})$ and λ^{l}, λ^{α} and λ^{β} are the eigenvalues of Cov. U is the used as rotation matrix.

Transformation is carried out as:

$$I = Tsrc.Rsrc.Ssrc.Stgt.Rtgt.Ttgt.Itgt \qquad (2)$$

where $I = (l;\alpha;\beta;1)^{T}$ and $Itgt = (ltgt\ ;\alpha tgt\ ;\beta tgt\ ;1)^{T}$ denote the homogeneous coordinates of pixel points in lαβ space for the result and target images respectively; *Tsrc*, *Ttgt* , *Rsrc*, *Rtgt* , *Ssrc* and *Stgt* denote the matrices of translation, rotation and scaling derived from the source and target images separately. Here is their definition:

$$Tsrc = \begin{vmatrix} 1 & 0 & 0 & t_{src}^{l} \\ 0 & 1 & 0 & t_{src}^{\alpha} \\ 0 & 0 & 1 & t_{src}^{\beta} \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad Ttgt = \begin{vmatrix} 1 & 0 & 0 & t_{tgt}^{l} \\ 0 & 1 & 0 & t_{tgt}^{\alpha} \\ 0 & 0 & 1 & t_{tgt}^{\beta} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$R_{src} = U_{src};\ R_{tgt} = U_{tgt}^{-1} \qquad (3)$$

$$S_{src} = \begin{vmatrix} S_{src}^{l} & 0 & 0 & 0 \\ 0 & S_{src}^{\alpha} & 0 & 0 \\ 0 & 0 & S_{src}^{\beta} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad S_{tgt} = \begin{vmatrix} S_{src}^{l} & 0 & 0 & 0 \\ 0 & S_{src}^{\alpha} & 0 & 0 \\ 0 & 0 & S_{src}^{\beta} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Where, t_{src}^{l}= lsrcmean, t_{src}^{α}= αsrcmean, t_{src}^{β}= βsrcmean; t_{tgt}^{l}=- ltgtmean, t_{tgt}^{α}= -αtgtmean, t_{tgt}^{β}= -βtgtmean ;$S_{src}^{l} = \lambda_{src}^{l}$, $S_{src}^{\alpha} = \lambda_{src}^{\alpha}$, $S_{src}^{\beta} = \lambda_{src}^{\beta}$; $S_{tgt}^{l} = 1/\lambda_{tgt}^{l}$, $S_{tgt}^{\alpha} = 1/\lambda_{tgt}^{\alpha}$, $S_{tgt}^{\beta} = 1/\lambda_{tgt}^{\beta}$

The subscripts of src and tgt indicate source image and target image, respectively. Finally the result is converted back to RGB color space.

IV. Global color transfer (second algorithm)

In this algorithm some aspects of the distribution of data points in *l*αβ space is transferred between images.

Mean and standard deviation along each of the three axes is computed for both source image & target image.
First the mean is subtracted from data points:

$$l^* = l - \langle l \rangle$$
$$\alpha^* = \alpha - \langle \alpha \rangle \quad (4)$$
$$\beta^* = \beta - \langle \beta \rangle$$

Data points of target image are scaled by factors determined by the respective standard deviations:

$$l' = \frac{\sigma_t^{\,l}}{\sigma_s^{\,l}}\, l^*$$

$$\alpha' = \frac{\sigma_t^{\,\alpha}}{\sigma_s^{\,\alpha}}\, \alpha^* \quad (5)$$

$$\beta' = \frac{\sigma_t^{\,\beta}}{\sigma_s^{\,\beta}}\, \beta^*$$

After this transformation, the resulting data points have standard deviations that conform to the source image. Next, instead of adding the averages that previously subtracted, the averages computed for the source image is added. Finally the result is converted back to RGB color space.

V. LOCAL COLOR TRANSFER

Color transformation is carried out in lαβ color space. 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 color range used. The only limitation is that the whole region must be inside this object.

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:

$$\mu_c^R = \frac{1}{N_R} \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} c(i,j) \quad (6)$$

$$\sigma_c^R = \sqrt{\frac{1}{N_R - 1} \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \left(c(i,j) - E^R c\right)^2} \quad (7)$$

Where $N_R = (i_2 - i_1 + 1)(j_2 - j_1 + 1)$ is the number of pixels in the selected region R, $1 \le i_1 < i_2 \le H$ and $1 \le j_1 < j_2 \le W$ define the rectangular region R and $c(i, j)$ is a processed color channel of pixel (i,j). Maximum distance from the mean is calculated in both source image and target image for the selected color range. 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. 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 Data points of the selected color range in target image are scaled by a factor determined by maximum distance from the mean. After scaling the mean of the rectangular region in source image is added. Here we have three sliders to user to have better control over color transfer. First slider is used to decide the size of mask. Second slider is used to adjust the how much color should be transferred and third slider provides user exponential adjustment on color transfer. The formula used to calculate the final pixel's value is as follows:

nl = labColorMean[0] + ((labTarget[0] - labTargetMean[0]) * totalMaxDistance1 / totalMaxDistance);
na = labColorMean[1] + ((labTarget[1] - labTargetMean[1]) * totalMaxDistance1 / totalMaxDistance);
nb = labColorMean[2] + ((labTarget[2] - labTargetMean[2]) * totalMaxDistance1 / totalMaxDistance); …………(8)

Here nl , na and nb are temporary variables whereas totalMaxDistance1 and totalMaxDistance indicates the maximum distance of pixel from the selected color range for source image and target image respectively.

Next step is to calculate the new pixel value according to the slider values.

labTarget[0] = labTarget[0] + (nl - labTarget[0]) * slider2 / 100 * Math.pow(slider3, (Euclidian / totalMaxDistance));
labTarget[1] = labTarget[1] + (na - labTarget[1]) * slider2 / 100 * Math.pow(slider3, (Euclidian / totalMaxDistance));
labTarget[2] = labTarget[2] + (nb - labTarget[2]) * slider2 / 100 * Math.pow(slider3, (Euclidian / totalMaxDistance)); (9)

Equestion (9) shows the resulting pixel's value which can be obtained by adjusting the sliders value. Final step is to transfer the lαβ color space into RGB color space.

VI. EXPERIMENTS

We have implemented the proposed algorithms in JAVA (object oriented language). The proposed algorithms are tested on the set of images downloaded from the internet. It is found that both global color transfer algorithms give best result for the source and target images which are similar in composition. The result of first global color transfer algorithm shows that not only the mood of source image is transferred to target image but the object's color of target image is recolored according to source image's object's color. The result of second global color transfer shows that it retains object's color in target image. It only transfers the appearance of source image to target image. The proposed local color transfer algorithm is tested on a set of images of different compositions. Our experiment included the cases of 1) using an image fragment or a single color as color source 2) correction of both luminance and chrominance or chrominance only; 3) using images of similar or different composition for the case of using an image fragment as color source. The time required for the local color transfer with the images of size 960x720 is 28seconds with core i3 processor. It preserves natural look of images and the color range which is selected by user modified rest of the image remains unaffected. The proposed algorithm gives best result with any size of images.

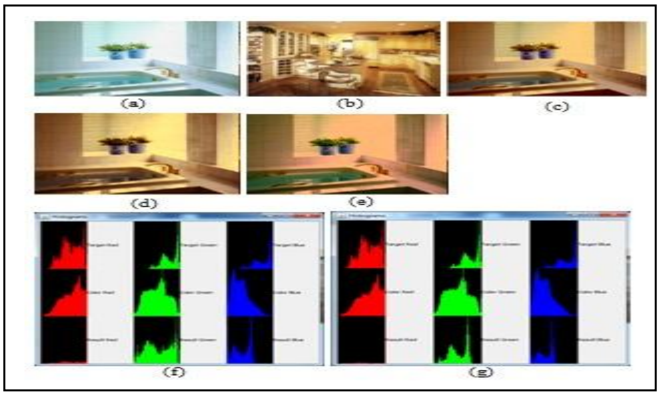

Figure 3. Comparison of resulting images; (a) target image, (b) source image, (c) resulting image of global color transfer[9], (d) resulting image of the global color transfer using algorithm first , (e) resulting image of the global color transfer using algorithm second,(f) Histogram comparison of images a,b and d, (g) Histogram comparison of images a,b and e.

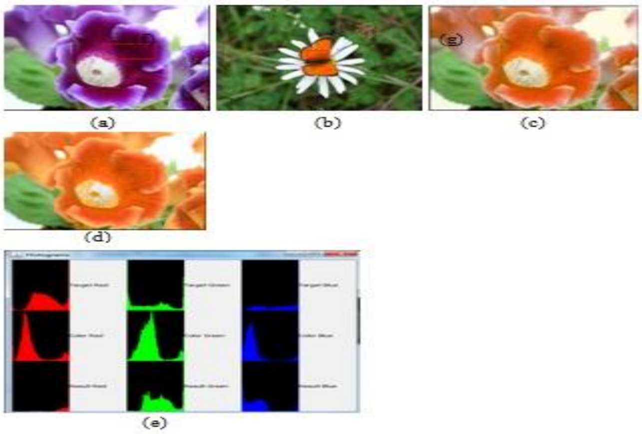

Figure 4. Comparison of resulting images; (a) target image, (b) source image, (c) resulting image of local color transfer[8], (d) resulting image of the local color transfer using proposed algorithm , (e) Histogram comparison of images a,b and d.

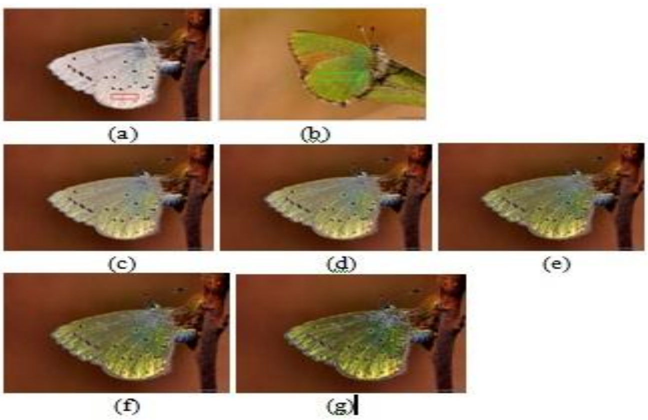

Figure 5. Result for local color transfer; (a) target image, (b) source image, (c)(d)(e)(f)(g) resulting images at various sliders values.

VII. LIMITATIONS

We have implemented both global as well as local color transfer algorithms. The main limitation of our work is our first global color transfer algorithm gives best result for monotonic images only & second global color transfer gives best result when source and target images are similar in composition. If images with multiple tones are used for global color transfer then our algorithm will not perform well.

VIII. CONCLUSION

Our global color transfer algorithms are the extension of Reinhard's. It is very hard to evaluate results objectively, the evaluation is really subjective. Our proposed global color transfer algorithms work well for the images with similar compositions. Our local color transfer algorithm allows user to correct an object of interest at an image saving one from the trouble of selecting it precisely. Also it allows user to obtain a desired color transfer just by adjusting the sliders. Only the selected object’s color range is changed preserving other color intact and achieving natural look of the result for wide variety of input images. Our proposed algorithms work with all real size images.

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.

[9] Xuezhong Xiao and Lizhuang Ma,”Color Transfer in correlated color space”, *ACM International Conference on Virtual Reality Continuum and Its Applications VRCIA 2006.*