`colortrans`
Color Transfer Algorithms

Daniel Steinberg

March 19, 2022

This document describes the color transfer algorithms that are implemented in the `colortrans` Python library[1].

There are an assortment of algorithms for transferring colors from one image to another. The image to be transformed—while retaining its qualitative appearance—is referred to as the *source* and the image providing the colors is referred to as the *target* or *reference* (it may be helpful to think of the source image as the *source* of content and the target image as having the *target* colors).

Global color transfer techniques transform the source image so that global color statistics of the transformed image match those from the reference image. Local color transfer approaches utilize regional correspondences between images for a more fine-grained transfer. More recently, a deep learning based technique has been developed for transferring colors in a way that considers semantic relationships across images [2].

We describe an assortment of global color transfer algorithms below. Some of these techniques were proposed along with extensions for local color transfer, which we omit from consideration. Without additional processing, a high-quality transfer using global techniques may require that the source and target images are similar in composition.

1 Reinhard

Reinhard et al. [7] develops a technique that transforms a source image in such a way that its channel means and standard deviations in $l\alpha\beta$ color space [8] match the associated statistics from the target image. An RGB image can be represented by a dataset $\boldsymbol{X}$, a matrix of pixel values where each row corresponds to an image position and columns correspond to the three RGB color channels. The following transformation converts an image from RGB to $l\alpha\beta$, with log denoting the element-wise base-10 logarithm:

$$\boldsymbol{X}^{l\alpha\beta} = \log\left(\boldsymbol{X}\begin{bmatrix}0.3811 & 0.1967 & 0.0241\\ 0.5783 & 0.7244 & 0.1288\\ 0.0402 & 0.0782 & 0.8444\end{bmatrix}\right)\begin{bmatrix}0.5774 & 0.4082 & 0.7071\\ 0.5774 & 0.4082 & -0.7071\\ 0.5774 & -0.8165 & 0.0000\end{bmatrix} \tag{1}$$

With both the source image s and target image t converted to $l\alpha\beta$, values for each channel from the source image are shifted by the associated channel c mean μ_s^c:

$$l^* = l - \mu_s^l \qquad \alpha^* = \alpha - \mu_s^\alpha \qquad \beta^* = \beta - \mu_s^\beta \tag{2}$$

The shifted values are scaled such that the standard deviations match those of the target image, with σ_i^c denoting the channel c standard deviation of image i:

$$l' = \frac{\sigma_t^l}{\sigma_s^l}\, l^* \qquad \alpha' = \frac{\sigma_t^\alpha}{\sigma_s^\alpha}\, \alpha^* \qquad \beta' = \frac{\sigma_t^\beta}{\sigma_s^\beta}\, \beta^* \tag{3}$$

[1] https://github.com/dstein64/colortrans

Next, the zero-centered values are shifted by the associated target image channel c mean μ_t^c so that the updated channel means match the target image means:

$$l^\dagger = l' + \mu_t^l \qquad \alpha^\dagger = \alpha' + \mu_t^\alpha \qquad \beta^\dagger = \beta' + \mu_t^\beta \tag{4}$$

Lastly, the preceding shifted values are converted back to RGB with the following operation—the inverse of Equation 1—that converts an image from $l\alpha\beta$ to RGB, with antilog denoting the element-wise base-10 antilogarithm:

$$\boldsymbol{X} = \text{antilog}\left(\boldsymbol{X}^{l\alpha\beta}\begin{bmatrix}0.5774 & 0.5774 & 0.5774\\ 0.4082 & 0.4082 & -0.8165\\ 0.7071 & -0.7071 & 0.0000\end{bmatrix}\right)\begin{bmatrix}4.4679 & -1.2186 & 0.0497\\ -3.5873 & 2.3809 & -0.2439\\ 0.1193 & -0.1624 & 1.2045\end{bmatrix} \tag{5}$$

2 Linear Histogram Matching

Linear Histogram Matching (LHM) [3] processes the pixels in a source image s so that the resulting distribution of pixel intensities approaches that of a target image t. Rather than matching the histogram exactly, which could result in "dramatically changing the appearance of the image," a linear transformation is used to match the target distribution's mean and covariance. Given a source image represented by a dataset comprised of vectors (3-dimensional for RGB images), each point $\boldsymbol{x}$ is multiplied by matrix $\boldsymbol{\beta}$ followed by the addition of vector $\boldsymbol{\alpha}$:

$$\boldsymbol{x}^* = \boldsymbol{\beta}\boldsymbol{x} + \boldsymbol{\alpha} \tag{6}$$

$\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ are set so that the resulting image statistics match the associated values of the target image:

$$\boldsymbol{\beta} = \sqrt{\boldsymbol{\Sigma}_t}\sqrt{\boldsymbol{\Sigma}_s}^{-1} \qquad \boldsymbol{\alpha} = \boldsymbol{\mu}_t - \boldsymbol{\beta}\boldsymbol{\mu}_s \tag{7}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean vector and covariance matrix for image i. After substituting $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ in Equation 6, the transformation becomes:

$$\boldsymbol{x}^* = \sqrt{\boldsymbol{\Sigma}_t}\sqrt{\boldsymbol{\Sigma}_s}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_s) + \boldsymbol{\mu}_t \tag{8}$$

For the preceding formulas, $\sqrt{}$ is defined such that $\sqrt{\boldsymbol{A}}\sqrt{\boldsymbol{A}}^\top = \boldsymbol{A}$, whose solution is not unique. Under LHM, where $\boldsymbol{A}$ is a symmetric covariance matrix with eigendecomposition $\boldsymbol{A} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top$, the matrix square-root $\boldsymbol{B} = \sqrt{\boldsymbol{A}}$ is computed as:

$$\boldsymbol{B} = \boldsymbol{\Phi}\sqrt{\boldsymbol{\Lambda}}\boldsymbol{\Phi}^\top \tag{9}$$

where $\sqrt{\boldsymbol{\Lambda}}$ is the element-wise square-root of the diagonal eigenvalue matrix $\boldsymbol{\Lambda}$. That is, the square root of a matrix is constructed by replacing eigenvalues with their square roots in the matrix eigendecomposition.

3 Principal Component Color Matching

Like the other global color transfer algorithms, **Principal Component Color Matching** (PCCM) [4, 5, 6] transforms a source image so that its color statistics match those of a reference image, while preserving the qualitative appearance of the source image. To achieve this, the source image data is rotated and scaled in a way that the resulting principal components of the output image match those of the reference image. When color transfer is considered from the perspective of domain adaptation, PCCM can be viewed as an application of Correlation Alignment (CORAL) [9].

The PCCM technique is similar to the method proposed by Abadpour and Kasaei [1, sec. II-A], but rather than using PCA and its inverse operation as-is, PCCM also incorporates a scaling operation. This ensures that the principal components of the transformed image not only have the same direction ("axes matching" [4]) as the reference image principal components, but also the same magnitude ("variance matching" [4]).

Given source image s and target image t, both RGB images represented by sets of 3-dimensional vectors, each source data point $\boldsymbol{x}$ is transformed by $\boldsymbol{x}^* = \boldsymbol{A}_t \boldsymbol{S} \boldsymbol{A}_s^\top (\boldsymbol{x} - \boldsymbol{\mu}_s) + \boldsymbol{\mu}_t$, where $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_t$ are mean data points from the source and target images, respectively. The 3×3 matrices $\boldsymbol{A}_s$ and $\boldsymbol{A}_t$—which serve "axes matching"—consist of eigenvectors of the source and target image dataset covariance matrices, one eigenvector per column. Scaling matrix $\boldsymbol{S}$—which serves "variance matching"—is a 3×3 diagonal matrix whose entries are the square roots of eigenvalue ratios, where the eigenvalues correspond to the eigenvectors used for constructing $\boldsymbol{A}_s$ and $\boldsymbol{A}_t$.

PCCM as Whitening-Coloring. Here we formulate PCCM in terms of *whitening* and *coloring*. The whitening transformation decorrelates a dataset such that its resulting covariance matrix is the identity matrix. Given a dataset d (i.e., a set of 3-dimensional vectors for an RGB image) whose covariance matrix $\boldsymbol{\Sigma}_d$ eigendecomposition is $\boldsymbol{\Phi}_d \boldsymbol{\Lambda}_d \boldsymbol{\Phi}_d^\top$ and mean data point is $\boldsymbol{\mu}_d$, the following shows the corresponding whitening transformation f_d as a function of data point $\boldsymbol{x}$:

$$f_d(\boldsymbol{x}) = \boldsymbol{\Lambda}_d^{-1/2} \boldsymbol{\Phi}_d^\top (\boldsymbol{x} - \boldsymbol{\mu}_d) \tag{10}$$

where the diagonal matrix $\boldsymbol{A} = \boldsymbol{\Lambda}_d^{-1/2}$ is defined by $(\boldsymbol{A})_{ii} = 1/\sqrt{\lambda_i}$, with $\lambda_i = (\boldsymbol{\Lambda}_d)_{ii}$. The coloring operation is the inverse of whitening. The following formulation shows the coloring transformation g_d, for dataset d, as a function of a data point $\boldsymbol{x}$:

$$\begin{aligned} g_d(\boldsymbol{x}) &= f_d^{-1}(\boldsymbol{x}) && (11) \\ &= \boldsymbol{\Phi}_d \boldsymbol{\Lambda}_d^{1/2} \boldsymbol{x} + \boldsymbol{\mu}_d && (12) \end{aligned}$$

where the diagonal matrix $\boldsymbol{B} = \boldsymbol{\Lambda}^{1/2}$ is defined by $(\boldsymbol{B})_{ii} = \sqrt{\lambda_i}$.

Under this formulation of PCCM, the colors of target image t are transferred to source image s by the application of the image s whitening operation followed by image t's coloring transformation. Thus, the color is transferred for each data point $\boldsymbol{x}$ by:

$$\begin{aligned} \boldsymbol{x}^* &= g_t(f_s(\boldsymbol{x})) && (13) \\ &= \boldsymbol{\Phi}_t \boldsymbol{\Lambda}_t^{1/2} \boldsymbol{\Lambda}_s^{-1/2} \boldsymbol{\Phi}_s^\top (\boldsymbol{x} - \boldsymbol{\mu}_s) + \boldsymbol{\mu}_t && (14) \end{aligned}$$

References

[1] Arash Abadpour and Shohreh Kasaei. A fast and efficient fuzzy color transfer method. In *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004.*, pages 491–494, December 2004. doi: 10.1109/ISSPIT.2004.1433825. 3

[2] Mingming He, Jing Liao, Dongdong Chen, Lu Yuan, and Pedro V. Sander. Progressive Color Transfer With Dense Semantic Correspondences. *ACM Transactions on Graphics*, 38(2):13:1–13:18, April 2019. ISSN 0730-0301. doi: 10.1145/3292482. 1

[3] Aaron Hertzmann. *Algorithms for Rendering in Artistic Styles*. Ph.D., New York University, May 2001. 2

[4] Hiroaki Kotera. A scene-referred color transfer for pleasant imaging on display. In *IEEE International Conference on Image Processing 2005*, volume 2, pages II–5, September 2005. doi: 10.1109/ICIP.2005.1529977. ISSN: 2381-8549. 2, 3

[5] Hiroaki Kotera, Hung-Shing Chen, and Tetsuro Morimoto. Object-to-object color mapping by image segmentation. In *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts IV*, volume 3648, pages 148–157. SPIE, December 1998. doi: 10.1117/12.334552. 2

[6] Hiroaki Kotera, Mitsunori Suzuki, and Hung-Shing Chen. Object-to-object color mapping by image segmentation. *Journal of Electronic Imaging*, 10(4):977, 2001. ISSN 10179909. doi: 10.1117/1.1407263. 2

[7] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, July 2001. ISSN 1558-1756. doi: 10.1109/38.946629. Conference Name: IEEE Computer Graphics and Applications. 1

[8] Daniel L. Ruderman, Thomas W. Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: implications for visual coding. *Journal of the Optical Society of America A*, 15(8):2036, August 1998. ISSN 1084-7529, 1520-8532. doi: 10.1364/JOSAA.15.002036. 1

[9] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2058–2065, Phoenix, Arizona, February 2016. AAAI Press. 2