Color transfer

CVFX @ NTHU

26 Feb 2015
Outline

Color transfer by histogram matching
Example: a color transfer method

Color grading in the movie industry

"Shooting a movie out of sequence and at scattered locations results in images of varying color values. The process of color timing balances the hues, provides continuity, and evokes specific moods through enhancement or manipulation of colors." – Homing Beacon
Histogram equalization on a grayscale image

\begin{verbatim}
I = imread('a.jpg');
hist(I(:,), [0.5:1:254.5]);
J = histeq(I);
hist(J(:,), [0.5:1:254.5]);
\end{verbatim}
Line up and regroup

\[ \text{Line graph} \Rightarrow \text{graph} \Rightarrow \text{equalized (uniform)} \]
Histogram equalization via cumulative distribution function (c.d.f.)

Use the rescaled c.d.f. as a mapping function to set the new intensity value of every pixel.

\[ x' = C_X(x) \]
In histogram equalization, we expect the output image to have a uniform histogram.

What if we want it to have a specific distribution (other than uniform)?
Histogram matching

CXY(x)  CXY−1(CX(x))

Source  Result
Histogram matching

![c.d.f. of result](image1)

![c.d.f. of target](image2)

![result](image3)

![output](image4)
function f = pdf_transfer1D(pX,pY)
    nbins = max(size(pX));

    PX = cumsum(pX);
    PX = PX/PX(end);

    PY = cumsum(pY);
    PY = PY/PY(end);
% inversion
small_damping = (0:nbins+1)/nbins*1e-3;
PX = [0 PX nbins] + small_damping;
PY = [0 PY nbins] + small_damping;

f = interp1(PY,...
[0 ((0:nbins-1)+1e-16) (nbins+1e-10)], ..., PX, 'linear');
f = f(2:end-1);
end
Color distribution in RGB space

\[ x = (r, g, b)^T \]
Matching color distributions in RGB space
The idea

- Convert N-dimensional histogram matching into one-dimensional histogram matching
- Project the data onto one-dimensional space
The algorithm

1: **Initialization** of the data set source \( x \) and target \( y \).
   For example in color transfer, \( x_j = (r_j, g_j, b_j)^T \).
   \( k \leftarrow 0, \ x^{(0)} \leftarrow x \)

2: **repeat**

3:   take a rotation matrix \( R \) and rotate the samples:
   \( x_r \leftarrow Rx^{(k)} \) and \( y_r \leftarrow Ry^{(k)} \)

4:   project the samples on each axis \( i \) to get the marginals
   \( f_i \) and \( g_i \)

5:   for each axis \( i \), find the 1D transformation \( t_i \) that matches
   the marginals \( f_i \) and \( g_i \)

6:   remap the samples \( x_r \) according to the 1D transformations

7:   rotate back the samples: \( x^{(k+1)} \leftarrow R^{-1}x_r \)

8:   \( k \leftarrow k + 1 \)

9: **until** convergence on all marginals for every possible rotation
A\times x = B \quad x = A \backslash B; \quad ( ) \times = ( )

%%% match the marginals
for i=1:nb_projs
    f\{i\} = pdf\_transfer\_1D(p0R\{i\}, p1R\{i\});

    scale = (length(f\{i\})-1)/(datamax(i)-datamin(i));

    D0R\_(i,:) = interp1(0:length(f\{i\})-1, f\{i\}', ... (D0R(i,:) - datamin(i))*scale)/scale + datamin(i);
end

D0 = relaxation * R \backslash (D0R\_ - D0R) + D0;
Dissimilarity between two histograms

- **Kullback-Leibler distance**

\[ D_{KL}(p\|q) = \sum_i p_i \log \left( \frac{p_i}{q_i} \right), \quad \sum_i p_i = 1, \quad \sum_i q_i = 1 \]

- If two histograms are identical \( D_{KL}(p\|q) = 0 \)
How to check whether two distributions are identical?

- For every rotation of the axis, the projections (or marginals) of $f$ match the projections of $g$. 
How to check whether two distributions are identical?

- Every projection of $f$ matches the projection of $g$
How to choose rotation matrices?
A direct test on the package provided by the authors

Note by the author: The grain reducer technique is not provided here.

http://www.mee.tcd.ie/~sigmedia/Research/ColourGrading
"Color transfer between images" (Reinhard et al.)

\[
\begin{bmatrix}
L \\
M \\
S
\end{bmatrix} =
\begin{bmatrix}
0.3811 & 0.5783 & 0.0402 \\
0.1967 & 0.7244 & 0.0782 \\
0.0241 & 0.1288 & 0.8444
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

\[L = \log L, \quad M = \log M, \quad S = \log S\] (log10 in Matlab)

\[
\begin{bmatrix}
l \\
\alpha \\
\beta
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{\sqrt{3}} & 0 & 0 \\
0 & \frac{1}{\sqrt{6}} & 0 \\
0 & 0 & \frac{1}{\sqrt{2}}
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & -2 \\
1 & -1 & 0
\end{bmatrix}
\begin{bmatrix}
L \\
M \\
S
\end{bmatrix}
\]

\[I_s \rightarrow I_t\]

This example is to modify \( I_t \) according to \( I_s \)

\[
l_t' = \frac{\sigma_s}{\sigma_t} (l_t - \langle l_t \rangle) + \langle l_s \rangle
\]

\[
\alpha_t' = \frac{\sigma_s}{\sigma_t} (\alpha_t - \langle \alpha_t \rangle) + \langle \alpha_s \rangle
\]

\[
\beta_t' = \frac{\sigma_s}{\sigma_t} (\beta_t - \langle \beta_t \rangle) + \langle \beta_s \rangle
\]
“Color transfer between images” (Reinhard et al.)

\[
\begin{bmatrix}
L \\
M \\
S
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & -1 \\
1 & -2 & 0
\end{bmatrix} \begin{bmatrix}
\frac{\sqrt{3}}{3} & 0 & 0 \\
0 & \frac{\sqrt{6}}{6} & 0 \\
0 & 0 & \frac{\sqrt{2}}{2}
\end{bmatrix} \begin{bmatrix}
l \\
\alpha \\
\beta
\end{bmatrix}
\]

\[
L = 10^L, \quad M = 10^M, \quad S = 10^S
\]

derived from matrix inversion

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} = \begin{bmatrix}
4.4679 & -3.5873 & 0.1193 \\
-1.2186 & 2.3809 & -0.1624 \\
0.0497 & -0.2439 & 1.2045
\end{bmatrix} \begin{bmatrix}
L \\
M \\
S
\end{bmatrix}
\]
Another application

“Bayesian correction of image intensity with spatial consideration,” Jia, Sun, Tang, and Shum

Fast Shutter Speed

Slow Shutter Speed

Poor color

Good color but blurry
Another application

“Bayesian correction of image intensity with spatial consideration,” Jia, Sun, Tang, and Shum

After color mapping

Ground truth
Another application

Data-driven Hallucination of Different Times of Day from a Single Outdoor Photo

Yichang Shih
MIT CSAIL
Sylvain Paris
Adobe
Frédo Durand
MIT CSAIL
William T. Freeman
MIT CSAIL

Input image at “blue hour” (just after sunset)
A database of time-lapse videos
Hallucinate at night

Figure 1: Given a single input image (courtesy of Ken Cheng), our approach hallucinates the same scene at a different time of day, e.g., from blue hour (just after sunset) to night in the above example. Our approach uses a database of time-lapse videos to infer the transformation for hallucinating a new time of day. First, we find a time-lapse video with a scene that resembles the input. Then, we locate a frame at the same time of day as the input and another frame at the desired output time. Finally, we introduce a novel example-based color transfer technique based on local affine transforms. We demonstrate that our method produces a plausible image at a different time of day.

Abstract

We introduce “time hallucination”: synthesizing a plausible image at a different time of day from an input image. This challenging task often requires dramatically altering the color appearance of the picture. In this paper, we introduce the first data-driven approach to automatically creating a plausible-looking photo that appears as

Links: DL PDF

1 Introduction

Time of day and lighting conditions are critical for outdoor photography (e.g. [Caputo 2005] chapter “Time of Day”). Photographers often seek software to help them achieve the perfect light of...
Questions

- How to do histogram equalization?
- How to do one-dimensional histogram matching?
- How to do N-dimensional histogram matching?
- How to measure the dissimilarity between two histograms?
- Other applications?