Fast Image and Video Colorization
Using Chrominace Blending

IEEE Transactions on Imaging Processing
vol. 15, no. 5, May 2006
Liron Yatziv and Guillermo Sapiro

School of Electrical Engineering and Computer Science
Kyungpook National Univ.
Abstract

◆ Colorization
  – Coloring a gray scale image or video
  – Transforming luminance channel into RGB channels
  – Problem
    • Requiring some amount of human interaction or external information
    • Intensive computational cost

◆ Proposed method
  – Computationally simple, effective, fast method
  – Interactively getting the desired results
  – Possible extensions
    • Changing a existing color of image and the underlying luminance
Introduction

◆ Characteristics of conventional colorizations
  – Assumption
    • Geometry of the grayscale information provides the geometry of the image
  – Main shortcoming
    • Intensive computational cost

◆ Proposed method
  – Propagation method with less computational cost
  – Extension to video without additional method
  – Based on the color blending
  – Other effects such as recolorization
Fast colorization framework

Environmental of colorization

- $YCbCr$ color space
- Definition in the region

$$Y(x, y, \tau) : \Omega \times [0, T] \rightarrow \mathbb{R}^+$$

- Given monochromatic image ($T = 0$) or video ($T > 0$) defined on a region $\Omega$

$$Cb(x, y, \tau) : \Omega \times [0, T] \rightarrow \mathbb{R}^+$$

$$Cr(x, y, \tau) : \Omega \times [0, T] \rightarrow \mathbb{R}^+$$

- Given chromatic image ($T = 0$) or video ($T > 0$) defined on a region $\Omega$ (goal of proposed method)

- Input

- Values of chrominance channels provide by users in a region $\Omega_c \in \Omega$ (scribbles)
Process

- Letting \( s \) and \( t \) be two points in \( \Omega \)
- Letting \( C(p) : [0,1] \rightarrow \Omega \) to be a curve in \( \Omega \)
- \( C_{s,t} \); a curve connecting \( s \) and \( t \)
  - \( C(0) = s \) and \( C(1) = t \)
- Intrinsic (geodesic) distance
  - A measurement for integration of the difference of luminance between any two points

\[
d(s,t) := \min_{C_{s,t}} \left\| \nabla Y \cdot \dot{C}(p) \right\| dp
\] (1)
- Assumption
  - A close relationship between the basic geometry of the luminance and chrominance channels
    - Sharp luminance; a edge in the chrominance
    - Graduate luminance; not having an edge but a moderate change
  - A change in luminance; a related change in chrominance
    - The smaller \( d(s,t) \), the more similar chrominance
- Providing chrominances by users
  - Distance from a point \( t \) to be colored to a provided color \( s \) such as the same chrominance \( c \)

\[
d_c(t) := \min_{\forall s \in \Omega_c: \text{chrominance}(s) = c} d(s,t)
\]
Idea for colorization

- Computing $Cb$ and $Cr$ components of a point $t$ to be colored

$$
\text{Chrominance}(t) \leftarrow \frac{\sum_{\Omega \in \text{chrominace}(\Omega_c)} W(d_c(t))c}{\sum_{\Omega \in \text{chrominace}(\Omega_c)} W(d_c(t))}
$$

(3)

where \text{chrominace}(\Omega_c) stands for all the difference unique chrominance in the set $\Omega_c$.

$W(\cdot)$ is a function of the intrinsic distance that translates it into a blending weight

1) $\lim_{r \to 0} W(r) = \infty$;
2) $\lim_{r \to \infty} W(r) = 0$;
3) $\lim_{r \to \infty} W(r + r_0)/W(r) = 1, r_0$ a constant

Two or more chrominance sources close by
- Weighting factor

\[ W(r) = r^{-b} \]  

where \( b \) is the blending factor, \( 1 \leq b \leq 6 \)

Defining how smooth is the chrominance transition

◆ Performance

- Satisfactory colorization for human perception
  - Blending two or three chrominances
    - No needing a lot of memories
Recolorization and Extensions

- Recolorization
  - Assumption
    - Homogeneity in the chrominance indicates homogeneity in the new chrominance
  - Intrinsic distance
    - \( Cb \) and \( Cr \) channels
  - Generalization
    - Measurement medium
      \[
      M(x, y, \tau) : \Omega \times [0, T) \rightarrow \mathbb{R}
      \]
      - Any channel the input image and a mix of the channels
    - Blending medium (Both in colorization and recolorization)
      - Actually blended data
        \[
        B(x, y, \tau) : \Omega \times [0, T) \rightarrow \mathbb{R}
        \]
Colorization results

Fig. 1. Still-image colorization examples. Given a grayscale image, (left) the user marks chrominance scribbles, and (right) our algorithm provides a colorized image. The image size/run time from top to bottom are 230 x 345/less than 0.42 s, 256 x 256/less than 0.36 s, and 600 x 450/less than 1.73 s.
Fig. 2. Comparison of visual quality with the technique proposed in [17]. (a) Given grayscale image; (b) the user marks chrominance scribbles (the selected scribbles are obtained from the work in [17]); (c) our algorithm results with CPU run-time of 0.76 s; (d) Levin et al. approach with CPU run-time of 24 s using their supplied fast implementation based on multigrid solver (611 s are needed for their standard Matlab implementation). We observe the same quality at a significantly reduced computational cost.
Fig. 3. Comparison of visual quality with the technique proposed in [17]. (a) Given grayscale image (400 x 240); (b) the user marks chrominance scribbles; (c) our algorithm results with CPU run time of 1.20 s; (d) Levin et al. approach with CPU run time of 22.8 s using their supplied fast implementation of multigrid solver; (e) Levin et al. approach using a slower exact Matlab least squares solver also provided by the authors.
Fig. 4. Testing the robustness of the proposed algorithm. The first row shows an example where different sets of scribbles are used, obtaining visually identical results. The original monochromatic image is shown first, followed by the first set of scribbles and the resulting colorized result. This is followed by a second set of scribbles and the corresponding colorized results. Note how the two results are virtually identical. The next row repeats the same experiment for a different image. Next, we show (third row) how the result is evolving with the addition of new scribbles by the user. The original image is presented first, followed by the set of scribbles labeled by their order of application. The third figure shows the result when only the two scribbles labeled 1 are used. Then, scribbled 2 is added to improve the results on the top of the ball, obtaining the fourth figure. Finally, the scribble labeled 3 was added to keep the shadow gray instead of green, and the result is provided in the last figure.
Fig. 5. Colorization with palettes. After the scribble is propagated, the gray value is used to key the palette. The original image is followed by the colored one in the first row. The second row shows the palettes for the mountain and the sky, respectively.
Fig. 6. Animated video colorization example. (a) Given the ten-frame grayscale sequence, (b) our algorithm provides a colorized video. (c) The first and last frame of the colorized video are enlarged to show the color content. (d) To colorize the ten frames, all that was needed were a few chrominance scribbles on a single frame. The user marked chrominance scribbles on the seventh frame of this sequence. The size of each frame is 312 x 264, and the algorithm total run time for the whole sequence is 6.8 s. The movie can be found on the web site for this project.
Fig. 7. Video colorization example. (a) Given a 21 frame grayscale sequence, (b) our algorithm provides a colorized video. (c) The first and last frame of the colorized video are enlarged to show the color content. (d) To colorize the 21 frames, all that was needed are a few chrominance scribbles on a single frame. The user marked chrominance scribbles on the eleventh frame of this sequence. The size of each frame is $320 \times 240$, and the algorithm total run time for the whole project.
Fig. 8. Recolorization example. (a) Original color image, (b) scribbles, and (c) image after recolorization (note the pants, glasses, and window). The white scribbles represent a “mask,” used to keep the original colors.
Fig. 9. Recolorization example using the Cr channel for measurement (M) rather than the intensity channel. (a) Original color image; (b) intensity channel of the image. It can be clearly seen that the intensity image does not contain significant structural information on the red rainbow strip (this is quite a unique example of this effect). On the other hand, both the Cb and Cr do change significantly between the stripes and can, therefore, be used for recolorization. (c) Recolored image, the red stripe of the rainbow was replaced by a purple one. image.
Fig. 10. Example of the generalization of our method to other image processing effects. (a) The original color image. Our goal is to change the color of the yellow car into a darker color. (b) We define the blending medium by roughly marking areas we do not wish to change in white and areas we do want to change in black; we do so by placing scribbles or by just marking whole areas. (c) Using our colorization method, we propagate the markings (white and black colors) and get a grayscale matte (we only keep the blending channel). With the matte, it is possible to apply an effect to the original image with a magnitude proportional to the gray level of the matte. In this case, we chose to change the brightness. (d) Image after applying the darkening. Note that the intensity only changed in the desired parts of the image. The darkening is done simply by subtracting the gray-level matte from the intensity channel, where white means no change and black is the maximum preselected change. (e) It is possible to further process the image using the same matte and (f) demonstrate this by similarly adding the gray-level matte to the Cb and Cr channels, respectively.
Conclusion

◆ Proposed method
  – A fast colorization algorithm for still images and video
  – Quality as good as conventional algorithms
  – Less performance time within a second
  – Less manual effort and interactively due to high speed
  – Leading to recolorization and changing the luminance
  – Understanding the best position to place the scribbles

◆ Application
  – Image compression
    • Selection of the simplest set of color scribbles