

Temporal Color Morphing

Xuezhong Xiao and Lizhuang Ma

Department of Computer Science and Engineering
Shanghai Jiao Tong University, China

Abstract. Many natural phenomena usually appear in the scenes of movies. They are characterized by color changing smoothly following the time's advancement, e.g. day breaking, leaves' resurgence, foliage greening from yellow or yellowing from green. They are difficult to be captured wholly because of the long time span. Traditional methods for making these types of scenes are inefficient or makeshifts.

In this paper, we propose a semi-automatic temporal color morphing technique for the simulation of smooth color-changing phenomena. Our temporal color morphing method is based on histogram manipulation—histogram warping and histogram-based color transformation. For the simplicity of the calculation, we convert the input of the source sequence and target image to $l\alpha\beta$ color space. This way we split the three-dimensional problem to three one-dimensional problems. The coordinates are quantized in $l\alpha\beta$ space with a limited error condition. For customization of the results by the user, we introduce a timing mechanism for controlling the velocities of color morphing. Finally, we apply our temporal color morphing technique to the video clips of several representative natural phenomena to demonstrate the effectiveness of our method.

1 Introduction

There are many scenes of natural phenomena appeared in movies, e.g. day breaking, leaves' resurgence, foliage greening from yellow or yellowing from green, which behave as some processes and are characterized by color changing smoothly following the time's advancement. We call them *temporal color morphing*. These phenomena may last several hours or even several months whereas in movies these scenes just span several minutes or seconds.

Traditionally film makers have two types of methods to catch these scenes. One is that photographers take the whole process and film cutters cut and edit it to be a sequence with appropriate length, and another is that they shoot the initiative and terminative parts of the whole process respectively and produce the scene transiting from the initiative part to the terminative part by the use of cross-dissolves. The first method is inefficient and impractical in some situations, e.g., foliage yellowing which could cost several months. The second method is a makeshift, it can not produce image sequences showing the evolving process of these natural phenomena and to some extent its results just give audience some mental hints.

In this paper, we propose a semi-automatic temporal color morphing technique for the simulation of smooth color-changing phenomena. We contend that there are three key points of the temporal color morphing algorithm:

- Availability: our algorithm needs input which can be gained easily. And it can produce plausible results. That is, the results look like the simulated natural phenomena visually.
- Adjustability: the algorithm should provide a way for users to impact the final results. Example properties are timing and target style selection.
- Efficiency: the algorithm acts on image sequences (or video clips), a several seconds' clip has over one hundred frames. So short processing times are very important.

Our algorithm takes an image sequence (called source sequence) and a static image (called target image) as inputs. The source sequence should contain the desired scene except the color alteration. So we commonly take the initiative or terminative part of the simulated phenomenon as the source sequence. Correspondingly, the target image can be the terminative or initiative photo of the simulated scene or other scene's photo which should have the target color characteristics. Obviously, we can conveniently capture the source sequence and target image. The algorithm then automatically creates the sequence with desired color alteration.

Our temporal color morphing technique is based on histogram manipulation in $l\alpha\beta$ color space. It is known that the histogram is an elaborate and useful image statistic which sums up the distribution of images' color. So we express the holistic effects of color alteration as histogram warping. After storing the intermediate states of histogram warping, a histogram-based color transformation algorithm is presented for generating the corresponding in-between frames. We also provide a parameter for users to control the color morphing speed.

Color histogram manipulation in RGB color space is a difficult task because there exist correlations between the three channels' values and any histogram manipulation must operate on the three components simultaneously. $l\alpha\beta$ color space is developed to minimize correlations between the three channels for natural scenes [1]. After transforming to $l\alpha\beta$ space, we can manipulate the three channels' histograms separately. That is, the three dimensional problem is converted into three one-dimensional problems. The complexity of the problem is reduced greatly. We also quantize $l\alpha\beta$ coordinates to improve the algorithm's performance under a limited error condition.

2 Related Work

Our work is about color processing. Many color processing methods have been published recently, e.g. about colorization, color-to-gray, color transfer and color harmonization.

Colorization [2, 3] is a technique for adding color to gray-scale images in order to increase the visual appeal of images. Welsh et al. describe a semi-automatic

example-based technique for colorizing a gray-scale image by transferring color from an example color image [2]. Levin et al.’s method colorizes gray-scale images according to the scribbles marked by users using optimization [3].

Color-to-gray is an inverse procedure with colorization and a dimension reduction problem. Gooch et al. introduces a salience-preserving algorithm to convert color images to gray-scale [4]. The algorithm works in a perceptually uniform CIE $L^*a^*b^*$ color space. They use chrominance and luminance differences to create gray-scale target differences between nearby image pixels, and selectively modulate the gray-scale representation by solving an optimization problem. Simultaneous research by Rasche et al. on the color-to-gray problem transforms color to gray-scale preserving image detail by maintaining distance ratios during the reduction process [5, 6]. They also extend it to aid color-deficient observers.

Reinhard et al. introduce a method for color transfer between images [7]. They convert pixel values in RGB space to Ruderman et al.’s perception-based color space $l\alpha\beta$ [1], and scale and shift $l\alpha\beta$ values to produce very believable results using simple statistics—the mean and standard deviations. Our method works in the $l\alpha\beta$ space and is based on statistics too. However, the statistics used in our method are histograms that are more detailed than mean and standard deviations and are suitable for our applications. Also, our algorithm processes video clips.

Wang and Huang [8] proposed a color transfer method for image sequences. Their algorithm takes a single input image and three target images as inputs, so their resultant sequences keep a static scene while it exhibits color alteration. Our algorithm produces the results in which the scene and color can be changing simultaneously (Fig. 3 and Fig. 6). Furthermore, our method is based on the histogram manipulation which is more elaborate than the color moments (Fig. 5).

Cohen-Or et al.’s color harmonization method detects the best harmonic scheme for the color of a given image and enhances the harmony among the color while remaining faithful to the original color [9].

From the term “morphing”, we naturally think of image morphing techniques. Image morphing is a powerful visual effects tool that creates a fluid transformation of one digital image into another [10]. It begins with feature correspondence, and then builds warp functions, and finally interpolates the positions and color to generate inbetween images [10]. Feature correspondence is a very important step for generating good effects, usually it is achieved by experienced animators. Our proposed color morphing is a semi-automatic method. The thing that users need to do is selecting the source sequence and target image for the input. Of course, users can also participate in the timing of color alteration. Another important point is that image morphing techniques cannot be used for simulating the natural phenomena which is our method’s task.

Histogram warping and histogram-based color transformation are the core of our method. The correlation between three color components is always a intractable problem for color histogram manipulation. Grundland and Dodgson [12] selects $L^*a''b''$ color space as their space for color histogram warping. But large numbers of experiments by Ruderman et al. [1], Reinhard et al. [7] and

Welsh et al. [2] indicate that $l\alpha\beta$ color space is an appropriate space for processing natural images. Grundland and Dodgson [11, 12] interpolate histograms using piecewise rational quadratic splines and warp the interpolating splines to find the color correspondence. For the simplification of calculation, our method generates intermediate states by directly interpolating between two histograms.

Our histogram-based color transformation is very similar to histogram specification. Color histogram specification is a difficult problem because of three correlated dimensions. Gonzalez advised matching the color intensities and leaving the color themselves unchanged [13]. But his method results in dissatisfying results. Our method changes the three dimensional problem as three one-dimensional problems through converting RGB values to the $l\alpha\beta$ color space.

3 The Algorithm

In this section, we describe in detail the temporal color morphing algorithm. Our goal is to construct a mapping function $f(\cdot)$ which defines a transformation that leads the source sequence's color from the beginning to the end to morph to the target image's characteristics.

The inputs (the source sequence and the target image) are RGB values. Firstly, they are converted to the decorrelated $l\alpha\beta$ color space. Then, there are two phases of processing performed on the three color components separately—histogram warping and histogram-based color transformation. After this process, we must transform the results back to RGB space. The color space transformation procedures between RGB and $l\alpha\beta$ follow directly from Reinhard et al. [7] except that the $l\alpha\beta$ data need to be quantized for higher performance. Figure 1 shows the framework of our algorithm.

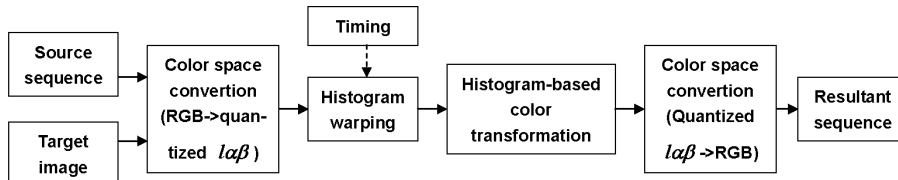


Fig. 1. The framework of temporally color morphing.

The pixel values in an image may be viewed as random variables. We use random variables s , r and z to denote the source sequence, resultant sequence and target image, respectively, and their definition domain is $[C_{min}, C_{max}]$. Then histograms are represented by corresponding variables' probability density function that is one of the most fundamental descriptors of a random variable. Let $g_t(s)$, $p_t(r)$ and $h(z)$ denote the probability density functions of random variables s , r and z , respectively, while t denotes the t th frame in the source or

resultant sequence. So the aforementioned mapping function can be formulized as $r = f(s, z)$.

3.1 Histogram Warping

For higher performance, we calculate the in-between histograms using the linear interpolation method. The speed of color alteration is controlled by a timing curve. Here is the formula of our histogram warping algorithm:

$$\int_{C_{min}}^x p_t(r)dr = \int_{C_{min}}^x g_1(s)ds + \left(\int_{C_{min}}^x h(z)dz - \int_{C_{min}}^x g_1(s)ds \right) \cdot \frac{\int_0^t v(\omega)d\omega}{\int_0^N v(\omega)d\omega} \quad (1)$$

while N is the total number of frames in source sequence, $v(t)$ denotes the timing curve representing the speed of color changing and is the parameter for users' control. We define four velocity functions including three basic functions and their combination (Fig. 2).

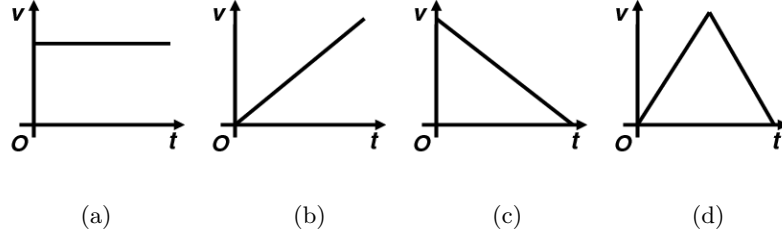


Fig. 2. Timing curves: (a) uniform speed, (b) uniform acceleration, (c) uniform deceleration, (d) combined curve.

3.2 Histogram-based Color Transformation

After histogram warping, we have two groups of histograms, one is the source sequence's and another is for the result sequence. They are $g_t(s)$ and $p_t(r)$, while $t = 1, 2, \dots, N$, and $p_1(r) = g_1(s)$, $p_N(r) = h(z)$. Each pair of $g_t(s)$ and $p_t(r)$ is the input of histogram-based color transformation algorithm.

Here, the goal is to find a transformation function which maps s to r . We assume that the transformation function satisfies a condition: it is single-valued and monotonically increasing in s 's definition domain $[C_{min}, C_{max}]$.

A basic result from elementary probability theory is that, if $g_t(s)$ and the transformation function are known and satisfy the above single-valued and monotonically increasing condition, the probability density function $p_t(r)$ can be obtained using a rather simple formula [13]:

$$p_t(r) = g_t(s) \left| \frac{ds}{dr} \right|$$

We sort the data set in ascending order, so we get the following equation:

$$\int_{C_{min}}^r p_t(\omega) d\omega = \int_{C_{min}}^s g_t(\omega) d\omega$$

Let $P_t(r)$ and $G_t(s)$ denote the source and result images' cumulative distribution density functions respectively, then we obtain the equation of histogram-based color transformation as follows:

$$r = P_t^{-1}[G_t(s)] \quad (2)$$

while

$$P_t(r) = \int_{C_{min}}^r p_t(\omega) d\omega$$

$$G_t(s) = \int_{C_{min}}^s g_t(\omega) d\omega$$

After merging Eq. (1) into Eq. (2), we obtain the mapping function for temporal color morphing as follows:

$$r = f(s, z) = P_t^{-1}[G_t(s)] \quad (3)$$

while

$$P_t(r) = G_1(s) + [H(z) - G_1(s)] \cdot k$$

$$H(z) = \int_{C_{min}}^z h(\omega) d\omega$$

$$k = \frac{\int_0^t v(\omega) d\omega}{\int_0^N v(\omega) d\omega}$$

The procedure we develop for temporal color morphing is summarized as follows:

Step 1: Defining a timing curve which could be one of the three basic curves or their arbitrary combination (Fig. 2).

Step 2: Calculating the cumulative distribution density functions $H(z)$, $G_t(s)$ and $P_t(r)$.

Step 3: Interpolating $P_t^{-1}(\cdot)$ for each $x = G_t(s)$ but not within $P_t(r)$ and build a look-up table $s \rightarrow P_t^{-1}(\cdot)$.

Step 4: Finding the value from the look-up table $s \rightarrow P_t^{-1}(\cdot)$ for every pixel in t -th frame.

Running through the four steps once, the algorithm generates one frame.

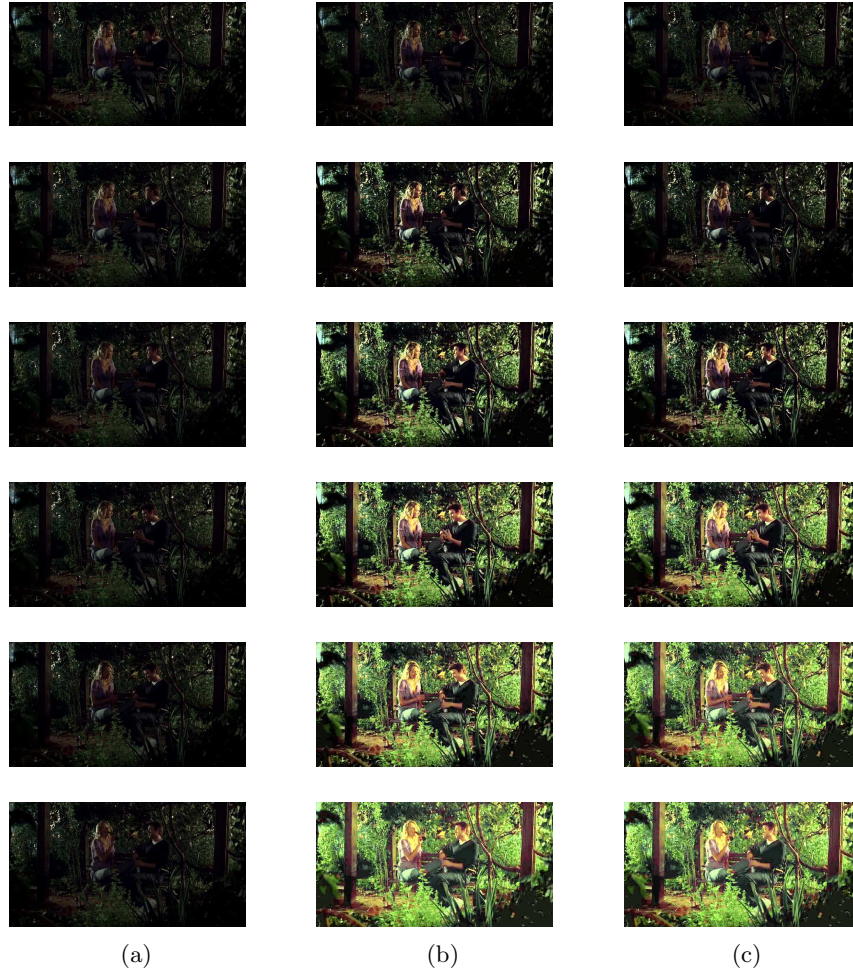


Fig. 3. A woman and a man chatting through the caliginous early morning to the midday (Frames No.1, 60, 120, 180, 240 and 300, from left to right). The left column (a) is from the source sequence, the middle column (b) is the effect adjusted by the uniform speed timing curve in Fig. 2(a), and the right column (c) is adjusted by the combined timing curve in Fig. 2(d).

3.3 Color Coordinate Quantization

Unlike the RGB color having 256 regular values along each axis, each component of $l\alpha\beta$ color has over ten thousand values. This results in Step 3 in the above procedure to be the bottleneck of the algorithm. So the quantization of color coordinates in $l\alpha\beta$ space is very important.

In order to accelerate the algorithm, we quantize the coordinates' values of $l\alpha\beta$ according to allowable errors in different applications. We quantize the coordinate vectors using LBG algorithm [14] and set the terminate condition as

$$d(\{R, G, B\}, \{R^*, G^*, B^*\}) < e \quad (4)$$

while e denotes the allowable error, $\{R, G, B\}$ and $\{R^*, G^*, B^*\}$ are the original coordinate in RGB space and the coordinate transformed from quantized $l\alpha\beta$ value, respectively. Namely, $\{R^*, G^*, B^*\}$ is obtained by the procedure of $\{l, \alpha, \beta\} = T(\{R, G, B\})$, $\{l^*, \alpha^*, \beta^*\} = Quantization(\{l, \alpha, \beta\})$ and $\{R, G, B\} = T^{-1}(\{l^*, \alpha^*, \beta^*\})$ while $T(\cdot)$ and $T^{-1}(\cdot)$ are the RGB to $l\alpha\beta$ conversion and its inverse and $Quantization(\cdot)$ denotes a LBG algorithm.

4 Results

We apply our temporal color morphing technique to the video clips of several representative natural phenomena to demonstrate the effectiveness of our method. Fig. 3, 6 and 7 exhibit the selected frames produced by this technique. Fig. 4 shows the target images used in the paper.

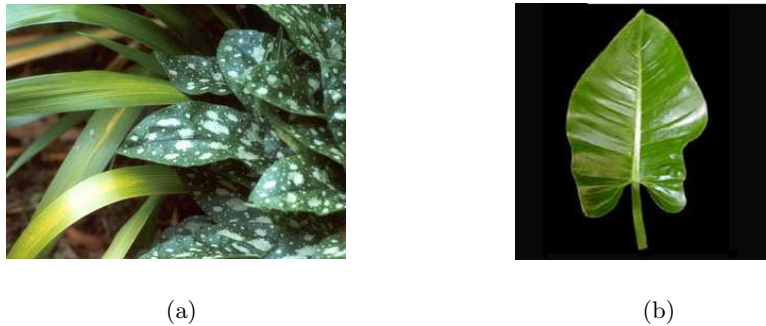


Fig. 4. The target images used in the paper. (a) Green leaves, used by Fig. 3; (b) Banana leaf, used by Fig. 6 and 7.

Fig. 3 shows a scene of chatting between a woman and a man through the caliginous early morning to the midday. The target image for them is Fig. 4(a). Fig. 3 also showcases the effects impacted by different timing curves. The left column is the selected frames from the source sequence, the middle column is

the effect by a uniform speed timing curve (Fig. 2(a)), and the right column is resulted by the combined timing curve (Fig. 2(d)). The other examples are adjusted by the uniform speed timing curve.

The swinging branch in Fig. 6 is morphing its color to the color characteristics of Fig. 4(b).

When the frames of the source sequence are the same, they are decayed to be a static image. Fig. 7 is this type of examples. A dried leaf is resurgent in Fig. 7 when its color moves forward to fit the characteristics of Fig. 4(b).

Our histogram-based color transformation can be taken solely as an algorithm for color transfer between images. Fig. 5 shows the results produced by Reinhard et al.'s method and ours. The result by our method is more accordant with the target style (b). Reinhard et al.'s method is more efficient. So the users must leverage the efficiency and the effect to select an appropriate method.



Fig. 5. Color transfer using our histogram-based color transformation and Reinhard et al.'s method: transferring (b)'s color into (a). (c) is the result by Reinhard's method and (d) is ours.

Our experiments all run on a computer with Pentium 4 CPU (2.80GHz) and 1G Bytes memory. The running time of our implementation ranges from 1 second to 3 seconds per frame. The results shown here were all obtained by using a quantized $l\alpha\beta$ color space. The allowable error e in Eq. 4 is 2 and the error for each component of RGB color returned from quantized $l\alpha\beta$ color is less than 1 (while $R, G, B \in [0, 255]$).



Fig. 6. Swinging branch is greening (Frames No. 1, 24, 48, 72, 96 and 120).



Fig. 7. The resurgence of a dried leaf (Frames No. 1, 30, 60 and 90).

5 Discussion And Conclusion

In this paper we present a temporal color morphing technique for producing the scenes of some natural phenomena in movies which are characterized by smooth color alteration along the time axis. While traditional methods accomplish this task by cutting and editing inefficiently or makeshifts, our approach generates plausible results and the needed inputs can be obtained conveniently.

This paper demonstrates that histogram warping can effectually represent the characteristics of temporal color morphing phenomena and that $l\alpha\beta$ color space's usefulness for color processing. We split the three-dimensional problem into three one-dimensional problems in $l\alpha\beta$ space and solve them more efficiently. We propose a quantized $l\alpha\beta$ color space under the condition of an allowable error.

Acknowledgments. This work is supported by National Science Fund for Creative Research Groups 60521002 and national natural science foundation of China (Grand No. 60573147 and No. 60173035).

References

1. Daniel L. Ruderman, Thomas W. Cronin, Chuan-chin Chiao: Statistics of Cone Responses to Natural Images: Implications for Visual Coding. *J. Optical Soc. of America*, vol 15, no. 8, 2036-2045.
2. Tomihisa Welsh and Michael Ashikhmin and Klaus Mueller: Transferring color to greyscale images. *ACM Trans. Graph*, 21(3), pp. 277-280, 2002.
3. Anat Levin, Dani Lischinski, Yair Weiss: Colorization using optimization. *ACM Trans. Graph*, 23(3), pp. 689-694, 2004.
4. A. A. Gooch, S. C. Olsen, J. Tumblin, B. Gooch: Color2Gray: salience-preserving color removal. *ToG*, 24(3), pp. 634-639, 2005.
5. Karl Rasche, Robert Geist, James Westall: Detail Preserving Reproduction of Color Images for Monochromats and Dichromats. *IEEE Computer Graphics and Applications*, 25(3), pp. 22-30, 2005.
6. Karl Rasche, Robert Geist, James Westall: Re-coloring Images for Gamuts of Lower Dimension. *Computer Graphics Forum*, 24(3), pp. 423-432, 2005.
7. Tomihisa Welsh, Michael Ashikhmin, Klaus Mueller: Transferring Color to Greyscale Images. *SIGGRAPH 2002 Conference Proceedings, Annual Conference Series*, pp. 277-280, ACM Press/ACM SIGGRAPH, 2002.
8. Chung-Ming Wang, Yao-Hsien Huang: A Novel Color Transfer Algorithm for Image Sequences. *J. Inf. Sci. Eng*, 20(6), pp. 1039-1056, 2004.
9. Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, Ying-Qing Xu: Color harmonization. *ACM Trans. Graph*, 25(3), pp. 624-630, 2006.
10. George Wolberg: Image morphing: a survey. *The Visual Computer*, Vol. 14, Number 8/9, pp. 360-372, 1998.
11. Mark Grundland, Neil A. Dodgson: Automatic Contrast Enhancement By Histogram Warping. *International Conference on Computer Vision and Graphics, Computational Imaging and Vision*, Vol. 32, pp. 293-300, Springer, 2004.
12. Mark Grundland, Neil A. Dodgson: Color histogram specification by histogram warping. *Color Imaging X: Processing, Hardcopy, and Applications (San Jose, USA, 17-20 January 2005)*. *Proceedings of SPIE*, vol. 5667, pp. 610-621, Society of Photo-Optical Instrumentation Engineers.

13. Y. Linde, A. Buzo, R. M. Gray: An algorithm for vector quantizer design. IEEE Trans. on Communications, Vol. COM-28, Number 1, pp. 84-95, January 1980.
14. Rafael C. Gonzalez and Richard E. Woods: Digital image processing, Second Edition. Prentice Hall, 2002.