A Novel Color Transfer Algorithm for Image Sequences

CHUNG-MING WANG AND YAO-HSIEN HUANG
Institute of Computer Science
National Chung Hsing University
Taichung, 402 Taiwan
E-mail: cmwang@cs.nchu.edu.tw

This paper presents a novel image sequence color transfer algorithm (ISCT). It is able to render an image sequence with color characteristics borrowed from three user-given target images. The input of this algorithm consists of a single input image (I_1) and three target images (T_1, T_2, T_3). The output of the algorithm is an image sequence $\{S_i\}$ with N images that embeds itself with color mood variations. A user selects necessary parameters, such as N and a color variation curve (CVC), through a user friendly interface we have developed. The algorithm completes the task in three steps. First, it executes a color transfer algorithm using the input image and three target images, producing three basis images (B_1, B_2, B_3). Their color features are thus borrowed from the three target images (T_1, T_2, T_3). Given the user-defined CVCs and three basis images generated in the first step, the ISCT algorithm then interpolates the mean and variance values for N images. Finally, referring to the input image and values interpolated in the second step, the algorithm automatically generates an image sequence. This is achieved in several seconds by applying the color transfer algorithm individually for every interpolated value. In addition, we have developed a user interface which makes it possible to view the rendered image sequence. Experimental results show that our algorithm swiftly produces an image sequence containing color variation with little user intervention. Given only three target images, the novel ISCT algorithm can demonstrate its ability to produce an image sequence with color mood variation as well as visually plausible effects. This algorithm is automatic, effective, and expeditious, and is appropriate for many applications.

Keywords: image sequences, color transfer algorithm, color space, color correction, image rendering

1. INTRODUCTION

Images are efficient media for information exchange and communication [1]. An image sequence consists of a series of images, each of which possesses slightly different features, also known as coherence characteristics [2]. As a result, image sequences provide plentiful information and cherish emotions. Not surprisingly, image sequences have been widely used in many applications, such as video editing and computer animation [1, 2].

Colors act as a significant factor in the human visual system and are almost indispensable for human perception [3-7]. A common task in image processing is to alter an image's color. A color transfer algorithm may be used to complete this task [3-5]. The algorithm uses simple statistical analysis to impose the color characteristics of the target image on the source image. As a result, believable output images are produced given

Received March 30, 2004; accepted June 30, 2004.
Communicated by Han-Chieh Chao.

suitable target images. Though a color transfer algorithm is available for static images, to our knowledge, no similar color transfer algorithms have been proposed for image sequences. We believe it is crucial to develop such an algorithm capable of producing image sequences with color mood variation. This algorithm, if available, would present at least three advantages. First, we could lower costs. Given a limited number of images, we could make use of the algorithm to render large image sequences. Second, it could expand the feasibility of colors. We could employ this algorithm to transform one type of image sequence into another, producing much different visual effects. Third, this algorithm could perform tasks that are presently not possible. With the algorithm, we could transform ordinary images into surrealistic ones.

There are four goals that must be achieved if we seek to develop a color transfer algorithm for image sequences. First, the resultant image sequences must possess smooth transitions in color mood variation. This implies that a viewer must perceive image sequences without noticing any possible artifacts. This requirement is rather difficult to meet since the human visual system is sophisticated and very sensitive to luminance as well as color changes. Second, the algorithm should be automatically executable. It should require as little user intervention as possible. Third, the algorithm must perform as efficiently as possible so that the results will be promptly available. Finally, users must be able to execute the algorithm easily and have easily view the results.

This paper presents a novel image sequence color transfer algorithm (ISCT). Given an input image, the proposed algorithm is able to render an image sequence in which the color characteristics are borrowed from three user-given target images. Once users have selected necessary parameters through the user interface, the algorithm generates the resultant image sequence promptly. The ISCT algorithm is effective because it renders image sequences with more visually plausible effects. It is efficient since it produces results within several seconds. It is automatic, requiring little user intervention. Finally, it is convenient since the user interface provides much freedom in visualizing the results.

This paper is organized as follows. Section 2 reviews related works. In section 3, we describe the ISCT algorithm in detail. Section 4 presents experimental results obtained using two test models. Finally, section 5 presents conclusions and highlights possible future works.

2. RELATED WORKS

This section reviews related works presented on color transfer in image sequences. Unfortunately, the image sequence research to date has mainly focused on image sequence analysis and applications that are not related to color transfer issues. This makes it difficult to survey related works in the literature on color transfer in image sequences. Instead, we will survey color transfer algorithms which are mainly designed for single images [3-5]. Since most color transfer algorithms for single images are based on a particular color space, the $\mathbf{l\alpha\beta}$ color space [8], we will begin our survey by first briefly describing the $\mathbf{l\alpha\beta}$ color space.

2.1 The $l\alpha\beta$ Color Space

A color space, or a color model, is a specification of a 3D color coordinate system and a visible subset in the coordinate system within which all colors in a particular color gamut lie [4]. The purpose of a color space is to allow convenient specification of colors within some color gamut. Different color spaces use different methods to delineate color attributes. Some typical color spaces include RGB, LMS, and CMY, etc. The LMS color space is of particular interest since it possesses the channel correlation property. Fig. 1 shows correlation diagrams between the L and M channels and the L and S channels, respectively. They plot from 1000 pixels chosen at random from the image data set. Clearly, these diagrams show a high degree of correlation and asymmetry. Thus, changes made in one color channel should affect values in the other channel.

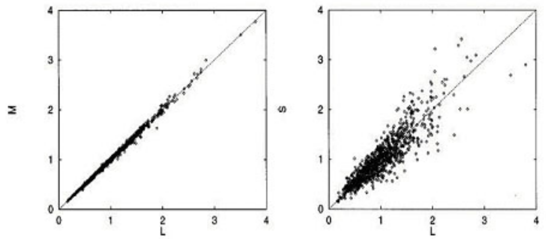

Fig. 1. Two diagrams showing the L-M (left) and L-S (right) distributions in the LMS color space [8].

Buchsbaum and Gottschalk discovered that orthogonal linear transformation could de-correlate a nature cone signal into 3 channels [9]. Ruderman continued along this line of research and presented a new color space, the **l**$\alpha\beta$ color space [8]. The **l**$\alpha\beta$ color space uses three channels to specify the color properties of an image. The **l** channel represents the luminance information, which is an achromatic channel. The chromatic α and β channels are roughly composed of yellow-blue and red-green opposing channels. Fig. 2 plots, in the logarithmic space, correlation diagrams between two of three channels [8]. Again, scatter plots of 1000 data points randomly selected are projected onto principal-axis pairs. The distributions are both highly elongated and compact in each of the three scatterplots. The data are distributed far more symmetrically than in the original, linear cone response space. Since the data are compact and dense, we can consider them as a linear superposition of independent sources, rather than just de-correlated sources. Unlike the LMS color space, the independent property in the **l**$\alpha\beta$ color space allows us to make changes in one color channel without affecting values in the others.

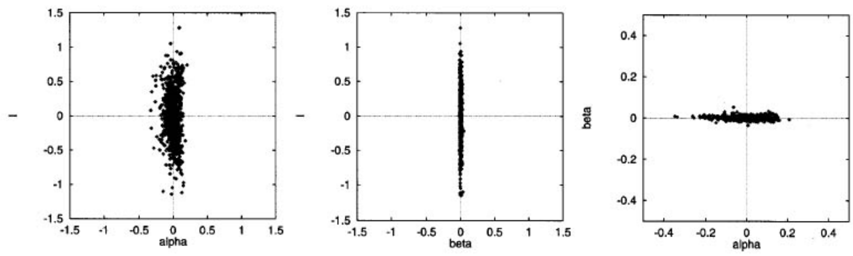

Fig. 2. Correlation diagrams in the logarithmic space for **l**-α (left), **l**-β (middle) and α-β (right) channels [8].

2.2 Color Transfer for a Single Image

Reinhard presented a color transfer algorithm for a single image [4]. Several other algorithms were later used to study similar issues; for example, transforming color images into greyscale images [5], speeding up the process [10], transforming colors based on basic color categories [11], and finally colorizing black-and-white cartoons [12].

The original color transfer algorithm uses statistical techniques to transfer the color mood in one source image to another target image [4]. First, both source and target RGB images are converted to the de-correlated **lαβ** color space for subsequent analysis. In particular, the source and target image are analyzed to obtain the respective mean and standard deviations at every axis. Then, we subtract the mean from the data points at each axis in the source image. We scale the data points using factors determined by the respective standard deviations. The resulting data points have standard deviations that conform to the target image following this transformation. We add the mean computed for the target image again. Finally, we convert the results back to the RGB color space, producing a final image on which color characteristics have been imposed. Eq. (1) highlights these transformations at each axis, where the subscript indicates whether a source or target image is used:

$$
\begin{aligned}
l &= \frac{\sigma^l_{Target}}{\sigma^l_{Source}}(l_{Source} - \mu^l_{Source}) + \mu^l_{Target}, \\
\alpha &= \frac{\sigma^\alpha_{Target}}{\sigma^\alpha_{Source}}(\alpha_{Source} - \mu^\alpha_{Source}) + \mu^\alpha_{Target}, \\
\beta &= \frac{\sigma^\beta_{Target}}{\sigma^\beta_{Source}}(\beta_{Source} - \mu^\beta_{Source}) + \mu^\beta_{Target}.
\end{aligned} \tag{1}
$$

The above color transfer algorithm has three advantages. First, it is simple, requiring only two images to produce the final result. Second, it is efficient, the computing time being less than a few minutes or even shorter. Third, it is semi-automatic, demanding very little user intervention during the color transfer process. However, this algorithm also demonstrates two disadvantages. First, it is a labor-intensive algorithm. Note that any image can be regarded as a target image once the source image is given. This implies that users will have to conduct trial-and-error tests to find a potential appropriate target image that can produce acceptable results. Second, this algorithm is subjective. It is difficult to measure the success of the resultant image since no absolute measurement metrics are currently available.

We think the original color transfer algorithm is not efficient for image sequence color transfer. This is because more target images are required in the case of an image sequence, and this in turn, eventually causes the original labor-intensive algorithm for a single image color transfer to deteriorate. To improve efficiency, we propose a novel color transfer algorithm, which uses only three images in order to accelerate the color transfer calculations. In addition, since the color transfer process has no absolute measurement metrics, the proposed algorithm must be flexible. Thus, we have developed a user interface for image color transfer to increase the flexibility of our algorithm. In the next section, we will discuss the proposed algorithm in details.

3. AN ALGORITHM FOR IMAGE SEQUENCES COLOR TRANSFER

In this section, we will describe the proposed algorithm for image sequence color transfer (ISCT algorithm). We will present an overview of the ISCT algorithm first before we describe the input, output, and transformation steps.

Fig. 3 shows an overview of the ISCT algorithm. The user is requested to provide an input image I_1 and three target images T_1, T_2 and T_3. In addition, the user has to provide an integer, N, indicating the numbers of image sequence that are to be rendered. Given the input image I_1 as the source image, we apply a color transfer algorithm to produce three basis images B_1, B_2, and B_3. We then define a color variation curve (CVC) which defines the characteristics of color transfer that a user intends to impose on the image sequence. The CVC has several properties. First, it contains the mean and variance of the three basis images B_1, B_2, and B_3 in the $\mathbf{l\alpha\beta}$ color space. Second, the two-end points of the CVC must be B_1 and B_3, or vice versa. Thirdly, the CVC must pass through B_2 at an arbitrary point given by the user. For example, when the CVC is linear and passes through B_2 exactly at the middle point, then the CVC may be a rising/fall line. Finally, the CVC is like a vector, which means that it can be represented with different types at each channel in the $\mathbf{l\alpha\beta}$ color space. Currently, our implementation utilizes an identical CVC for every channel in the $\mathbf{l\alpha\beta}$ color space. However, utilizing three CVCs in each channel is straightforward.

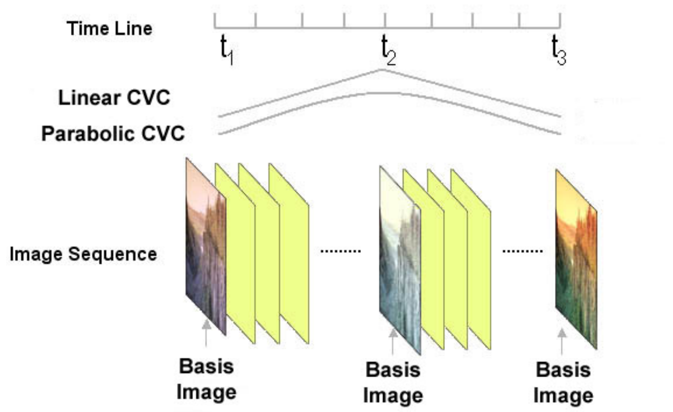

Fig. 3. An overview of the ISCT algorithm.

We divide operation of the ISCT algorithm into three steps, namely, forward color space conversion, the image sequence process, and finally image sequence animation. We will explain each step in the following sections.

3.1 Forward Color Space Conversion

The purpose of the forward color space conversion step is to transfer an image at the RGB color space to another perception-based color space **lαβ**. The term "forward" is based to the observation that the image to be transferred is normally in the RGB color space, while we normally perform the color transfer process in the **lαβ** color space. The need for conversion is due to the fact that when a typical image is represented in any of the popular, well-known color spaces, such as RGB, there will be correlations amongst the three channels [8]. Thus, this step de-correlates the color space by converting RGB signals into the **lαβ**. During transformation, the input image I_1 acts as a source image, while the three images that a user provides act as target images (T_1, T_2, T_3). Mathematically, this can be represented as shown in Eq. (2):

$$\begin{aligned} &I_{1(\mathrm{RGB})} \Rightarrow I_{1(\mathrm{l}\alpha\beta)}, \\ &T_{1(\mathrm{RGB})} \Rightarrow T_{1(\mathrm{l}\alpha\beta)}, \\ &T_{2(\mathrm{RGB})} \Rightarrow T_{2(\mathrm{l}\alpha\beta)}, \\ &T_{3(\mathrm{RGB})} \Rightarrow T_{3(\mathrm{l}\alpha\beta)}. \end{aligned} \tag{2}$$

The conversion process in this step is accomplished by means of matrix multiplications, as shown in Eqs. (3), (4) and (5). We have developed a function, **Forward-Color-Space-Conversion(),** to complete this task. Basically, this indicates that we first convert the RGB space into the LMS cone space. The data in this color space shows a great deal of skew, which we can largely eliminate by converting the data into the logarithmic space. Finally, we convert the intermediate results into the **lαβ** color space using two matrix multiplications:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{3}$$

$$\begin{aligned} &\boldsymbol{L} = \log L, \\ &\boldsymbol{M} = \log M, \\ &\boldsymbol{S} = \log S, \end{aligned} \tag{4}$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{L} \\ \boldsymbol{M} \\ \boldsymbol{S} \end{bmatrix}. \tag{5}$$

Once the images are in the **lαβ** color space, we apply the color transfer algorithm to transform the images into 3 basis images (B_1, B_2, B_3). Clearly, we have to calculate the mean and variance of these basis images in each channel, that is, (μ_i, σ_i), i = 1, 2, 3. Mathematically, this can be described as shown in Eq. (6).

$$\begin{aligned}\{I_{1(l\alpha\beta)}, T_{1(l\alpha\beta)}\} &\Rightarrow B_{1(l\alpha\beta)}\\ \{I_{2(l\alpha\beta)}, T_{2(l\alpha\beta)}\} &\Rightarrow B_{2(l\alpha\beta)}\\ \{I_{3(l\alpha\beta)}, T_{3(l\alpha\beta)}\} &\Rightarrow B_{3(l\alpha\beta)}\end{aligned} \tag{6}$$

In our experience, we have had to pay attention to overflow problems that may be encountered during mutual color space conversion. This means that when transferring pixel values in the **lαβ** color space back to the primitive color space, we have to guarantee that these values will remain within the range 0 to 255.

3.2 The Image Sequence Process

The purpose of this step is to render an image sequence that has a similar look and feel to 3 other basis images. In this step, we must refer to the numbers of image sequence (N) as well as the color variation curve (CVC), both of which are specified by the user. For the sake of clarity, we shall denote the image sequence that has been rendered as $S_{i(l\alpha\beta)}$ $(1 < i < N)$. Clearly, the image sequence to be rendered contains $N - 3$ images, excluding the 3 basis images that have been available. In addition, the first and the last image sequences correspond to the first and third basis images, respectively. Mathematically, this means that $S_{1(l\alpha\beta)} = B_{1(l\alpha\beta)}$ and $S_{N(l\alpha\beta)} = B_{3(l\alpha\beta)}$.

In the image sequence process, three functions are employed to complete the tasks. These functions are described in the following:

1. **Mean-Variance-Calculation():** This function calculates the means and variances for the three basis images in the **lαβ** color space. Obviously, the input of the function consists of three basis images (B_1, B_2, B_3), while the outputs are three vectors. Each vector has two components, representing the means and variances in each channel, respectively.

2. **Coefficients-Determination():** This function generates coefficients of the CVC, which will be used to generate the image sequence $(\{S_i\}, 1 < i < N)$. This function must refer to the means and variances being generated. It must also take into consideration the number of frames in the image sequence (N). For simplicity, we will consider the l channel only in the following discussion. Nevertheless, the two other channels can be employed in the same way.

The user can specify which types of CVC are to be employed. If the user chooses a single linear CVC, it has a general form, $y = ax + b$. Thus, we can obtain the above function to determine the coefficients (a and b) using two known points, $(x_1, y_1) = (0, B_1(\mu_1, \sigma_1))$ and $(x_2, y_2) = (N, B_3(\mu_3, \sigma_3))$. Alternatively, the user can select two linear CVCs, $y = ax + b$ and $y = cx + d$. In this case, the user has to specify an integer M $(1 < M < N)$ corresponding to B_2. Given these user-specified parameters, the coefficients a and b can be derived using two known points $(x_1, y_1) = (0, B_1(\mu_1, \sigma_1))$ and $(x_2, y_2) = (M, B_2(\mu_2, \sigma_2))$.

Similarly, the coefficients c and d can be derived using two other known points, $(x_3, y_3) = (M, B_2(\mu_2, \sigma_2))$ and $(x_4, y_4) = (N, B_3(\mu_3, \sigma_3))$. As previously noted, we utilize an identical CVC for every channel in the **lαβ** color space.

When the user chooses a non-linear CVC, such as a parabola, this CVC has the general form $y = ax^2 + bx + c$. We can assume that the CVC passes through three known data sets, $(x_1, y_1) = (0, B_1(\mu_1, \sigma_1))$, $(x_2, y_2) = (M, B_2(\mu_2, \sigma_2))$, and $(x_3, y_3) = (N, B_3(\mu_3, \sigma_3))$. Thus, the coefficients a, b, and c can be obtained with the expression shown in Eq. (7):

$$\begin{aligned} a &= \frac{(y_1 - y_2)(x_1 - x_3) - (y_1 - y_3)(x_1 - x_2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}, \\ b &= \frac{(y_1 - y_2)(x_1^2 - x_3^2) - (y_1 - y_3)(x_1^2 - x_2^2)}{(x_1 - x_2)(x_1 - x_3)(x_3 - x_2)}, \\ c &= \frac{y_1(x_1 - x_2)(x_1 - x_3)(x_2 - x_3) + x_1 x_3 (y_1 - y_2)(x_1 - x_3) - x_1 x_2 (y_1 - y_3)(x_1 - x_2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}. \end{aligned} \tag{7}$$

Similarly, we can apply the same approach to the two other channels and obtain the corresponding coefficients. Note that the coefficients a, b, and c in each channel are likely to be different since B_1, B_2, and B_3 may have different values in each channel.

3. **Image-Sequence-Rendering():** The objective of this function is to generate the image sequence $(\{S_i\}, 1 < i < N, i \neq M)$. Since the coefficients of the CVCs are now available, we can calculate the mean and variance corresponding to the i-th image, (μ_i, σ_i). For each i-th image, we apply the color transfer algorithm again, where the source image is always the input image, $I(\mu, \sigma)$, but the "target image" is a vector with the components μ_i and σ_i. Note that even though the "target" image does not exist, we can still perform color transfer so long as we provide the mean and variance corresponding to the target image. Clearly, the mean and variance corresponding to the target image have been calculated using the CVCs. The color transfer process for the i-th image is shown in Eq. (8), where the superscript indicates the corresponding channel, (l, α, β), and the subscript denotes the features of the i-th images:

$$\begin{aligned} S_i^l &= \frac{\sigma_i^l}{\sigma^l}(I^l - \mu^l) + \mu_i^l, \\ S_i^\alpha &= \frac{\sigma_i^\alpha}{\sigma^\alpha}(I^\alpha - \mu^\alpha) + \mu_i^\alpha \qquad (1 < i < N, i \neq M) \\ S_i^\beta &= \frac{\sigma_i^\beta}{\sigma^\beta}(I^\beta - \mu^\beta) + \mu_i^\beta. \end{aligned} \tag{8}$$

3.3 Image Sequence Animation

The image sequence completes two tasks. The first task is transferring the image sequence $\{S_{i(l\alpha\beta)}(1 < i < N)\}$ rendered by the image sequence process back to the primitive RGB color space. We can use the inverse equation to complete this task as shown in Eqs. (9), (10), and (11). Similar to forward color space conversion, we have developed a function, **Backward-Color-Space-Conversion()**, to complete this task. First, every image within the sequence is converted from the **lαβ** color space to the LMS color space using matrix multiplications. This is shown in Eq. (9). Then, the pixel values are amplified by a power of ten, as shown in Eq. (10). Finally, the pixel values are further converted to the RGB color space, as shown in Eq. (11). Once again, we have to solve possible overflow problems for pixel values when converting back into the primitive RGB color space:

$$\begin{bmatrix} \boldsymbol{L} \\ \boldsymbol{M} \\ \boldsymbol{S} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix}, \tag{9}$$

$$\begin{aligned} L &= 10^{L}, \\ M &= 10^{M}, \\ S &= 10^{S}, \end{aligned} \tag{10}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}. \tag{11}$$

The second objective in this step is to animate the image sequence $\{S_{i(\mathrm{RGB}),}\ 1 \le i \le N\}$. We believe that it is important to provide a smooth transition when animating the image sequence since this will result in a visually plausible appearance. This means that we have to take into consideration the number of frames in the image sequence N, which is given by the user. In our work, we have designed a user interface, and we request the user to adjust the frame rates (F) when visualizing the image sequence during the animation process (see "Frames" on the right of Fig. 4). The default value is 30, indicating a one-second animation. Clearly, the time duration for animating the image sequence is N/F seconds. This mechanism provides the user with complete freedom to control the desired time duration so that a smooth transition in the image sequence is possible.

4. EXPERIMENTAL RESULTS

We are aware that a demonstration video is available for presenting a paper entitled "transferring color to greyscale images" [5]. However, to our knowledge, color transfer for an image sequence has not been presented in the literature. As a result, no standard models were available to conduct experimental tests. Instead, we adopted our own two test models, the canyon model and the green tunnel model. They were retrieved from the Internet and used in the experiments. For the canyon model, we transferred the input image using the proposed ISCT algorithm to create the morning, midday, and evening canyon scenes (see Fig. 5 for details). As for the green tunnel model, the color was transferred on purpose to create yellow, green, and purple moods (see Fig. 9 for details). It was our intention that the resultant image sequence would have a surrealistic appearance, which is not possible without using color transfer techniques. In this section, we will first present the user interface we developed, and then we will present the animation result for the image sequences obtained using the two test models.

4.1 The User Interface

The platform for applying our algorithm was on a personal computer with a 1.6 GHz CPU, 256 MB memory, and the Windows operating system. Fig. 4 shows the user interface we developed. The user can use this interface to input basis images, and set up the color mood variation function and frame rates for animating the image sequence. We will divide our user interface into five parts to explain its functions.

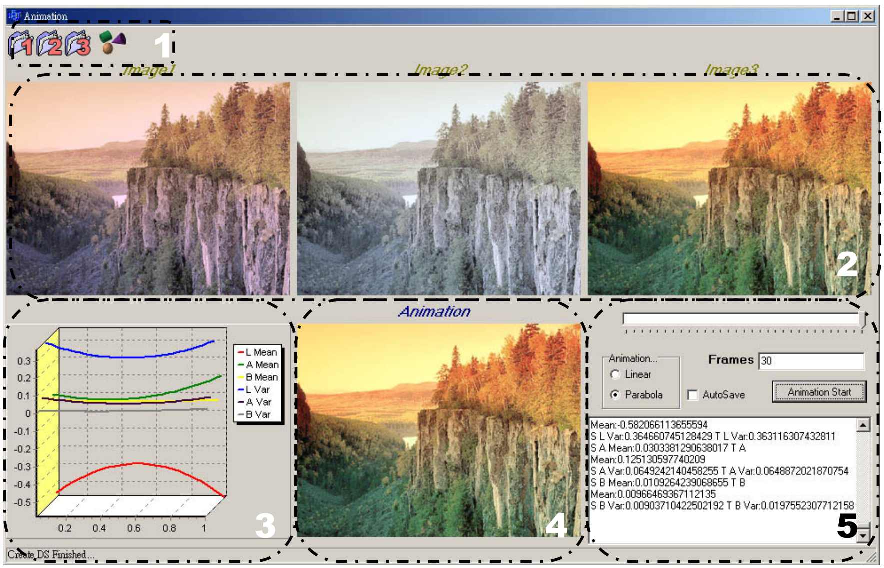

Fig. 4. The user interface we developed for the ISCT algorithm.

1. A tool bar (number 1) is provided at the top of the user interface with four buttons. The first three buttons (, ,) facilitate the input of three basis images. The final button () activates the forward color space conversion functions.
2. We display three basis images after the color transfer in the area (number 2). The order is from left to right.
3. This area (number 3) reveals the calculated means and variances for the resultant im-

age sequence. Six curves are displayed using different colors. These curves represent the means (**Mean**) and variances (**Var**) for the **l (L)**, α **(A),** and β **(B)** channels, respectively.

4. We show the animation results for the image sequence in this area (number 4).
5. This area (number 5) provides runtime statistics. The user can click on an animation style, is either a linear style or a parabola. In addition, the user can change the frame per second rate to control the display time duration. During the animation, a scroll bar (on the top) moves from left to right, providing frame cues for the user. Finally, the user can save the animation results as desired.

4.2 The Canyon Test Model

We transferred the canyon model (with a resolution of 320 × 240 pixels), given three target images, as shown in Fig. 5. The resultant images show the canyon in the morning, at midday, and in the evening. We select the number of frames in the image sequence at 30, where the first 15 frames showed the scene between morning and midday, and the other 15 frames showed the scene between midday and evening. We calculated 27 intermediate images (the means and variance in each channel), combined with three basis images for a total of 30 frames for the image sequence.

Fig. 5. The canyon input image (top), three target images (middle), and the resultant basis images after color transfer, showing morning, midday and evening moods (bottom).

Figs. 6 and 7 show the resultant image sequences generated by the ISCT algorithm. Fig. 6 shows the results obtained using the linear CVC, while Fig. 7 shows those obtained using the parabola CVC. Fig. 8 presents the calculated means and variances for the corresponding CVCs. Three basis images in Figs. 6 and 7 have black frame. Examining these figures, we observe that different CVCs created dissimilar color mode variations of the image sequence, as expected. In the canyon model, for example, the linear CVC changed colors gradually, while the parabolic CVC changed colors more quickly. In particular, the parabolic CVC gave the 20th frame an evening appearance, while the linear CVC retained the midday appearance. Another example is that the yellow color, which normally exists in the evening style, begins to be obvious in the 25th frame using a parabolic CVC. In contrast, when using a linear CVC, the yellow color becomes to be clear in the 21st frame. This indicates that the variation from the morning to evening appearance was stronger when the parabolic CVC was used.

Fig. 6. The animation results for the canyon model change color variation gradually. This sequence was rendered with a linear CVC.

Fig. 7. The canyon model, rendered with a parabolic CVC, demonstrates stronger color mood variations.

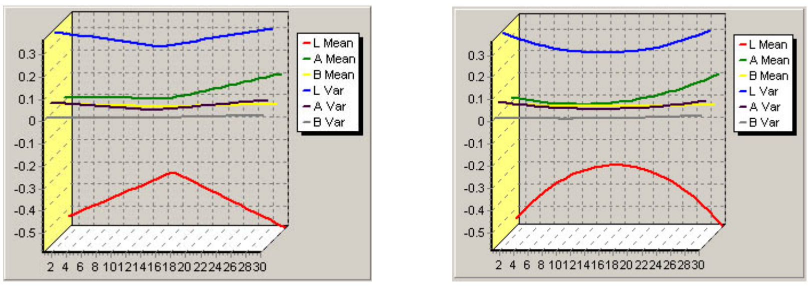

Fig. 8. The means and variances in each channel for the canyon model using a linear (left) and parabolic CVC (right).

The average execution time for the image sequence remained stable. It stayed in between 0.56 to 0.57 seconds for each frame, both for the linear and parabola CVC. We analyzed the difference between two consecutive frames in the image sequence using RMS measure metrics [13]. Table 1 presents the analysis results. These statistics match the trends of the mean values shown in Fig. 8. The mean values increased from the first frame to the middle frame and then decreased to the last frame. In addition, these statistics provide strong evidence that when we utilized a linear CVC, the variation between two consecutive images was mild in 30 frames of the image sequence. On the other hand, it was severe in the beginning and ending frames of the image sequence (the 1^{st} and 29^{th} frame), and was moderate in the middle (15^{th}) frame of the image sequence.

Table 1. The RMS values for the canyon test model using a linear CVC.

i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)
2	2.958501	12	3.194869	22	4.172726
3	3.101741	13	3.177671	23	4.125185
4	3.170869	14	3.207430	24	4.026994
5	3.147431	15	3.227489	25	3.886681
6	3.182105	16	4.472552	26	3.802374
7	3.152728	17	4.520751	27	3.741695
8	3.230110	18	4.539006	28	3.649553
9	3.192140	19	4.581950	29	3.591635
10	3.184265	20	4.460069	30	3.559347
11	3.221118	21	4.395413		

Table 2. The RMS values for the canyon test model using a parabolic CVC.

i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)
2	5.823286	12	1.571937	22	3.91657
3	5.616491	13	1.191709	23	4.410389
4	5.328909	14	0.911679	24	4.764669
5	4.917811	15	0.865514	25	5.036223
6	4.390394	16	1.075023	26	5.276625
7	3.853068	17	1.43243	27	5.482553
8	3.348807	18	1.864812	28	5.651544
9	2.872907	19	2.338442	29	5.800963
10	2.418215	20	2.844931	30	5.929304
11	1.986686	21	3.38894		

4.3 The Green Tunnel Test Model

Fig. 9 shows another test model: the green tunnel. For this model, the input image is shown at the top of the figure, and three basis images are shown in the bottom row to

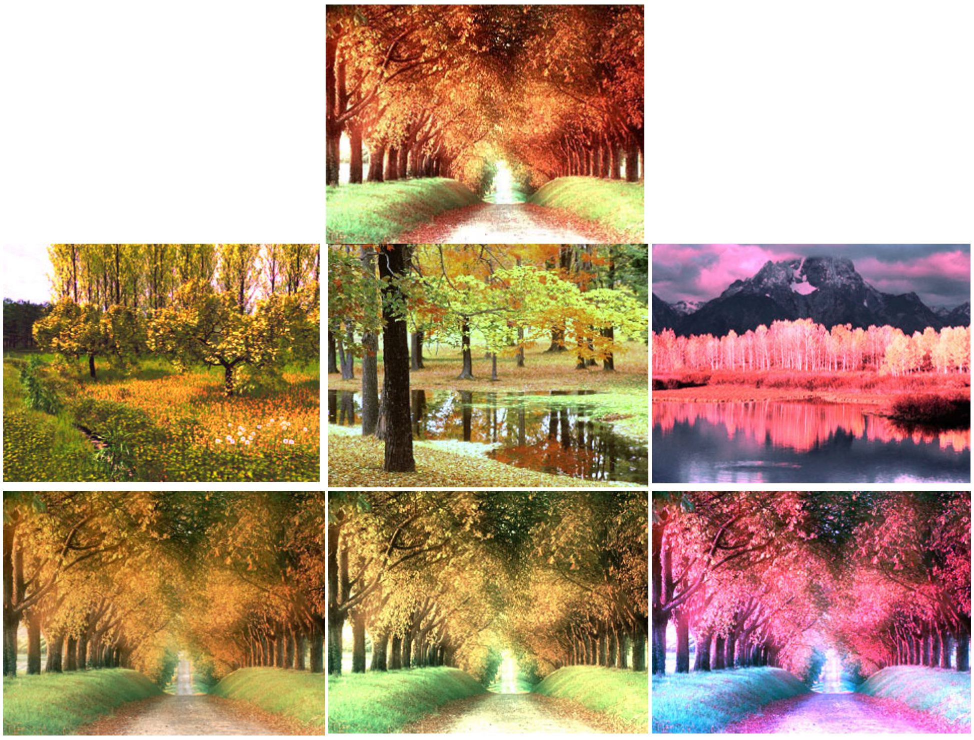

Fig. 9. The green tunnel input image (top), three target images (middle row), and the basis images with yellow, green and purple moods (bottom row).

demonstrate yellow, green, and purple color moods. Unlike the first canyon model, we selected an image sequence with 30 frames. Again, we calculated 27 intermediate images (the means and variance in each channel) along with the three basis images for a total of 30 frames in the image sequence.

Figs. 10 and 11 show the resultant image sequences generated by the ISCT algorithm using linear and parabolic CVCs, respectively. Again, three basis images have black contours. Examining these figures closely, we observe some trends that are different from those of the canyon model. The Linear CVC changed colors gradually in an uneven descendant manner, while the parabolic CVC again changed colors more quickly. In particular, the parabolic CVC causes color change in the 23rd frame, whereas the linear CVC keeps the color style until the red color mood starts to appear in the 20th frame. These images demonstrate a smooth color change.

The average execution time for the green tunnel model was similar to that for the canyon model; about 0.57 seconds was needed to calculate a single frame. Fig. 12 presents values of the mean and variance in each channel. Unlike the canyon model, however, the mean values in the luminous channel(l) increase gradually, while they remained constant in the beta channel(β), i.e., the red-green channel. However, the mean values decreased progressively in the alpha channel, that is, the achromatic channel. These changes in the three channels had two effects: luminance had the highest value in the 16th

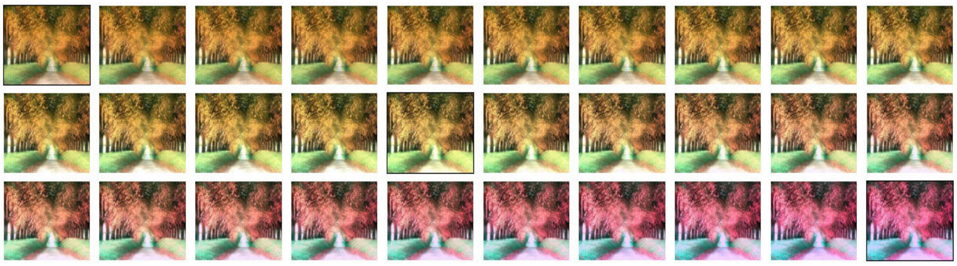

Fig. 10. The animation results for the green tunnel gradually change colors containing yellow styles to those holding red styles. This sequence was rendered with a linear CVC.

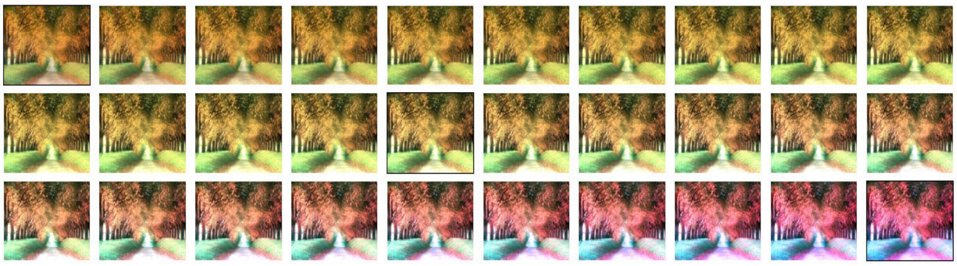

Fig. 11. The green tunnel model, rendered with a parabolic CVC, demonstrates strong color variation, where the red color mood is visible in the final 6 frames.

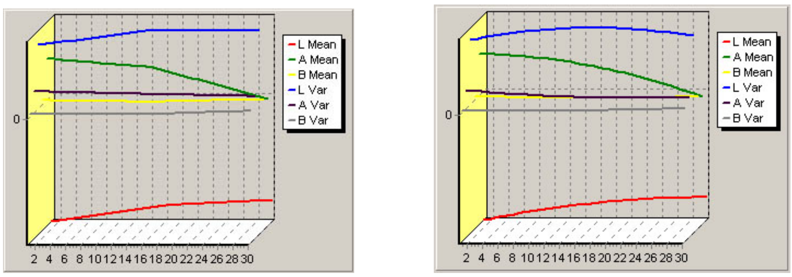

Fig. 12. The linear (left) curve and the parabola (right) curve of the green tunnel test models.

frame, and the yellow-blue signals declined as the red-green signals increased. Not surprisingly, red-green signals become more visible in the final frames.

Tables 3 and 4 present the RMS values for two consecutive frames in the image sequences obtained using the linear and parabolic CVCs, respectively. In the case of the linear CVC, the RMS values increased in general from the first to the last frame. However, when we used a parabolic CVC, the RMS values first gradually decreased to the minimum value at the 11th frame. The values then increased progressively to the maximum value at the last frame.

Examining the image sequences shown in Figs. 6, 7, 10, and 11, we observe that the color mood variation for the green tunnel model was less intense than that for the canyon

Table 3. The RMS values for the green tunnel test model obtained using a linear CVC.

i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)
2	2.956044	12	3.316249	22	3.775561
3	2.789507	13	3.167478	23	3.880527
4	2.806237	14	3.217171	24	3.790841
5	2.822103	15	3.155787	25	3.833524
6	3.019843	16	3.694782	26	3.837386
7	3.105009	17	3.754618	27	3.817879
8	3.138371	18	3.733139	28	3.838692
9	3.212190	19	3.832319	29	3.821272
10	3.188631	20	3.808452	30	3.887153
11	3.228999	21	3.841877		

Table 4. The RMS values for the green tunnel test model obtained using a parabolic CVC.

i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)	i	RMS($S_i - S_{(i-1)}$)
2	2.98029	12	1.878815	22	3.330789
3	2.85488	13	1.933376	23	3.51879
4	2.712035	14	2.028256	24	3.706099
5	2.556655	15	2.147701	25	3.874653
6	2.394541	16	2.288478	26	4.024879
7	2.237665	17	2.44277	27	4.149446
8	2.098009	18	2.610947	28	4.268107
9	1.984408	19	2.778395	29	4.38547
10	1.902816	20	2.958337	30	4.510337
11	1.869564	21	3.1422		

model. This may be due to the reaction of the human vision system. Note that the luminous channel in the canyon model changed more severely than that in the green tunnel model did. Note also that human beings have more the rod cells (for perceiving achromatic signals) than cone cells (for perceiving chromatic signals) [14]. The result is that we are more sensitive to the achromatic signal changes in the canyon model.

5. CONCLUSIONS AND FUTURE WORKS

Color is indispensable to the human visual system. An image sequence embedded with color mood variation provides an important dimension in visual communication. This article has presented a novel color transfer algorithm for producing image sequences. Given a single input image and three source images, the proposed ISCT algorithm renders an image sequence containing variations of color characteristics borrowed from three source images. The user selects necessary parameters through a friendly user interface we have developed, and the ISCT algorithm completes the task automatically in three steps within several seconds. Immediate feedback is provided for the user, who can easily view the animation results for a rendered image sequence.

We conclude that the ISCT algorithm is indeed feasible and effective, given some runtime statistics collected from the canyon and tunnel models. The algorithm generates a resultant image sequence containing reasonable and perceivable color mood variation. It is difficult for the viewer to see any artifacts, thanks to the linear and parabolic color variation curves that the user provides. Execution of the algorithm is automatic, requiring little user intervention, and performance is expeditious, taking only several seconds to render an image sequence of 30 frames. Adding different color variation curves will certainly produce image sequences with boundless, unlimited, and smooth features that are suitable in many applications.

In the future, we plan to investigate the use of different types of CVCs to render distinct image sequences. We intend to conduct human perception studies so that it will be possible to produce image sequences with more visually plausible effects. In addition, extending our algorithm to cope with special effects, such as motion blur or dynamic shadow effects, will be an interesting research direction. Finally, it is worthwhile to investigate possible application of the algorithm in computer animation to lower production costs and in the mobile computing environment for the purpose of entertainment.

REFERENCES

1. S. E. Umbaugh, *Computer Vision and Image Processing*, Prentice-Hall Inc., 1998.
2. T. R. Reed, *Digital Video Processing*, CRC Press, 2001.
3. G. R. Greenfield and D. H. House, "Image recoloring induced by palette color associations," *Journal of Winter School of Computer Graphics* (*WSCG*), Vol. 11, 2003, pp. 189-196.
4. E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, Vol. 21, 2001, pp. 34-41.
5. T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *Proceeding of ACM SIGGRAPH*, 2002, pp. 277-280.
6. J. M. Pinel and H. Nicolas, "Cast shadows detection on lambertian surfaces in video sequences," in *Proceedings SPIE*, Vol. 5150, *Visual Communications and Image Processing 2003*, 2003, pp. 378-384.
7. G. K. Wu and T. R. Reed, "Image sequence processing using spatiotemporal segmentation," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 9, 1999, pp. 798-807.
8. D. L. Ruderman, T. W. Cronin, and C. C. Chiao, "Statistics of cone responses to natural images: implications for visual coding," *Journal of Optical Society of America*, Vol. 15, 1998, pp. 2036-2045.
9. G. Buchsbaum and A. Gottschalk, "Trichromacy, opponent colour coding and optimum colour information transmission in the retina," in *Proceeding of Royal Society*, Ser. B 220**,** 1983, pp. 89-113.
10. G. D. Blasi and D. R. Recupero, "Fast colorization of gray images," *Eurographics Italian Chapter*, 2003.

11. Y. Chang, S. Saito, and M. Nakajima, "A framework for transfer colors based on the basic color categories," in *Proceedings of the Computer Graphics International*, 2003, pp. 176-181.
12. D. Sykora, J. Burianek, and J. Zara, "Unsupervised colorization of black-and white cartoons," To be appeared in *the 3rd International Symposium on Non-Photorealistic Animation and Rendering*, Annecy, France, 2004.
13. L. G. Shapiro and G. C. Stockman, *Computer Vision*, Prentice Hall, Upper Saddle River, New Jersey, 2001.
14. J. A. Ferwerda, "Elements of early vision for computer graphics," *IEEE Computer Graphics and Application*, Vol. 21, 2001, pp. 22-33.

Chung-Ming Wang (王宗銘) is an Associate Professor at the Institute of Computer Science, National Chung Hsing University, Taiwan. Wang received the B.S. degree from National Chung Hsing University and since then he has worked in industry for several years. He received his Ph.D. degree in Computer Science and Engineering from the University of Leeds, the United Kingdom, in 1992. Dr. Wang has won several Dragon Thesis Awards, funded by Acer Computers. His research interests include computer graphics, virtual reality, multimedia, and computer networks. Dr. Wang is a member of ACM, IEEE and Eurographics.

Yao-Hsien Huang (黃耀賢) received his B.S. degree in Computer Science and Information Engineering from Tamkang University and his M.S. degree in Computer Science from National Chung Hsing University, Taiwan, in 1990 and 1997, respectively. Now, he is working toward his Ph.D. degree at the Institute of Computer Science, National Chung Hsing University, Taiwan. His research work focuses on color transfer algorithms and image processing in computer graphics.