A New Algorithm for Adding Color to Video or Animation Clips

[1, 2]Zhigeng Pan [1]Zhaohua Dong [1]Mingmin Zhang

(1) State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310027, Zhejiang, China

(2) VRDM, Nanjing University, Nanjing, 210093, Jiangsu, China

{zgpan, dzh, zmm@cad.zju.edu.cn}

ABSTRACT

Colorizing grayscale video is a useful technique in scientific computing visualization and entertainment. In this paper, we introduce a method to transfer color from a reference image to the whole video. Although the general problem of adding chromatic values to a grayscale image has no exact, objective solution, the current approach attempts to free user from laborious work, except that the user may contribute their skill to reference image. Rather than choosing RGB colors from a palette to color individual components, we resort to anther color space, called lαβ, which minimizes correlation between channels for many natural scenes. Taking advantage of the correlation between two conjoint frames of video, we track the object and assign the color of it in the preceding frame to that of the posterior one. Experimental results show the algorithm works quite well.

Keywords

color transfer, image processing, color space, multimedia

1. INTRODUCTION

Adding color to grayscale video will increase the visual appeal of images such as black and white movies or scientific video. Moreover, the information content of some medical material can be perceptually enhanced with color by exploiting variations in chromaticity as well as luminance.

Studying the method of colorizing a grayscale image will provide some cue for the task. There exist a number of techniques for colorizing in information visualization. For example, Gonzalez and Wintz[6] describe a simple approach for pseudocoloring grayscale images. Pseudocoloring is a common technique for adding color to grayscale images such as X-ray, MRI, scanning electron microcopy and other imaging modalities in which color information does not exist. This method is also named as " image enhancement" technique, for it can be used to enhance the detectability of detail within the image. In fact, pseudocoloring is a transformation T, such that, $C(x, y) = T(f(x, y))$ where $f(x, y)$ is the original grayscale image and $C(x, y)$ is the resulting color vector for the three RGB color channels. A typical example of this method is the application of a color list being mapped to the grayscale image where a single, global color vector is assigned to the appropriate luminance. However, the results may have some perceptual distortions, because the color does not accord to the luminance. Another frequently adopted technique will render perfect results in principle, but requires much work and skill of user. By that means, image is divided into some segments and each segment is assigned with corresponding color. Apparently, this method does not appeal to common users, since it requires lots of interaction, experience and rendering time. For some complex objects such as hair and fur, patience and skill play the important role in picking up their shape, although some image processing software will help to find some shapes roughly.

In 2002, Welsh et al. [12] proposed a method to transfer color to grayscale image on Siggraph conference. They employed the lαβ color space, which was first introduced by Ruderman et al. [9] in 1998. Their algorithm transfers the entire color mood of the source to the target image by matching luminance and texture information between images. It chooses to transfer only chromatic information and retain the original luminance values of the target. Further, allowing the user to match areas of the two images with rectangular swatches to enhance the procedure.

Naturally, colorization of video can be automated using the colorization procedure described above. Every frame in the video sequence can be colorized using the same technique as is used in the single frame. Apparently, this method works for video colorization, regardless of processing time and human labor. Assuming a video clip lasts 10 seconds, whose frame rate is 20 frames per second, and it costs a minute to colorize one frame, thus the whole processing time of the video is 200 minutes. Furthermore, much interaction is necessary in order to avoid some stain. It takes quite long time to process the whole video clip. Fortunately, we find that commonly the same objects between adjacent frames move slightly. Therefore, when processing the posterior frame, we can take the preceding one as the reference image and assign the chromatic information from reference image to unmoving objects directly. For moving objects, trace the track of them and assign the corresponding chromatic information to them. As the experiment results show, this algorithm reduces the processing time and needs no user interaction with satisfactory color video clips.

The remainder of the paper is organized as follows. In Section 2, we discuss the lαβ color space and its advantage. Then, we describe our video colorizing algorithm in Section 3. Section 4 will show results of the algorithm and comparison of our algorithm and Welsh's image colorization extension. Conclusion will be given in Section 5.

2. lab COLOR SPACE

As Ruderman et al. mentioned, lαβ color space was developed to minimize correlation between the three coordinate axes of the color space. The color space provides three decorrelated, principal channels corresponding to an achromatic luminance channel l and two chromatic channels α and β, which roughly correspond to yellow-blue and red-green opponent channels. Small changes in one channel impose minimal effect on values of other two. The decorrelation in the three channels is the reason the lαβ space is chosen in current algorithm. Thus, we can independently assign the chromatic channels α and β from color reference image to the target grayscale image without cross-channel artifacts. Following is the conversion from RGB to lαβ and from lαβ to RGB.

We can convert the image from RGB to LMS space using the following conversion:

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

$\mathbf{L} = \log L$

$\mathbf{M} = \log M$

$\mathbf{S} = \log S$

Then transform from **LMS** to RGB follows:

$$\begin{pmatrix} l \\ a \\ b \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{pmatrix}$$

After video color processing, which explained in the next section, we must transfer the result back to RGB to be displayed with the inversed operation from lαβ to RGB.

$$\begin{pmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} l \\ a \\ b \end{pmatrix}$$

$L = 10^{\mathbf{L}}$

$M = 10^{\mathbf{M}}$

$S = 10^{\mathbf{S}}$

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

3. ASSIGN COLOR TO VIDEO

In a video clip, there exists correlation between adjacent frames. In many cases, the background will keep unchanged while an object is moving. Given two neighboring frames, I_k and I_{k+1}, each pixel p_1 (x, y) in I_{k+1} is assumed as coming from a pixel $p_2(x + u, y + v)$ in I_k:

$I_{k+1}(x, y) = I_k(x + u, y + v)$

Where u describes the horizontal velocity of the pixel and v describes the vertical velocity of the pixel. They are spatially varying function over image.

Over the years, researchers have developed a number of techniques to estimate the function of u and v, namely flow field, many of which are compared and summarized by Barron et al. [1]. Two assumptions common to many of these techniques are that the color of a source and destination pixel should be similar, and the flow field should exhibit some amount of spatial coherence. Thus, the problem becomes one of optimizing a data term (color similarity) plus a regularization term (smooth flow).

One of the better-performing optical flow techniques is due to Black and Anandan [2]. In addition to estimating regularized flow, their technique employs robust statistics to avoid large errors caused by outliers and to allow for discontinuities in the flow field. Their method handles large motions using a multi-scale approach to flow estimation.

In our video colorization, we introduce a new algorithm, which works well for many video clips, although it is quite simple. As illustrated in Figure 1, through searching preceding frame, we can determine color of the corresponding pixel. The horizontal and vertical displacements of the pixel are within a range. In a frame, the match procedure of neighboring pixels will provide some cue to the rest pixels. The cue information includes moving distance and direction, which will accelerate the match procedure of the other pixels.

Figure 1. The moving track of adjacent frames in a video clip.

In our method, key frames are selected and preprocessed to be assigned with color value. Just like video compress technique, in-between frames can be processed with two nearby key frames. The more temporally near is the target frame to a key frame, the more similar is the target frame to the key frame. Say, there are $f+b$ frames between two key frames and the target frame is f frames to the first key frame and b frames to the second key frame. Thus the similar coefficient of the first key frame is $\frac{b}{(f+b)}$ and that of the second key frame is $\frac{f}{(f+b)}$. We have:

$$R = R_f * C_f + R_b * C_b$$

Where R: the target frame result, R_f : the result of forward colorization, C_f: the similar coefficient of the first key frame, R_b the result of backward colorization, C_b: the similar coefficient of the second key frame.

The forward colorization and backward colorization is identical except that they take foregoing key frame and latter key frame as reference respectively. The procedure of colorization is described as bellow:

Step1. Colorize the key frames with Welsh's algorithm or some image processing software.

Step2. Assign color to a grayscale frame, taking the nearest key frame as reference image.

Step2.1 Transform RGB vector of the reference image and target image to $l\alpha\beta$ vector for subsequent analysis.

Step2.2 Compare the illumination of the pixels at the same coordinate (x, y). If the difference is lower than a given threshold ε, the illuminate of the two pixels are regarded as unchanged, i.e. the pixel does not move. Under this circumstance, chromatic information $\alpha(x, y)$ and $\beta(x, y)$ of the pixel in the reference image are transformed to target image directly. If the illumination difference is larger than ε, we can trace the pixel and transfer the color of reference image to target one, keeping illumination unchanged. The trace algorithm is as following:

For the pixel (x, y) in the target image, first, a small window is taken as the searching area, encircling the pixel (x, y) in the reference image. Expand the searching area if there is no matching pixel for the pixel in the target image, namely their illumination difference is larger than ε. Otherwise, stop searching and the current pixel is the matching pixel. The procedure is illustrated in Figure 2, where the dashdotted arrow denotes the searching direction within searching area and the dotted arrow denotes the expanding of searching area.

Step 3. If there is no matching pixel in the whole reference image, the pixel in the target image is regarded as noise spot. We can synthesize its α and β value with bilinear interpolation of nearby pixels.

The threshold ε plays an important role in the matching procedure. If it is too small, normal pixels are regarded as noise spots. While it is too large, two pixels of different illumination will be deemed identical and thus we get the result image of poor quality. In our experiment, the ε equals to 0.01.

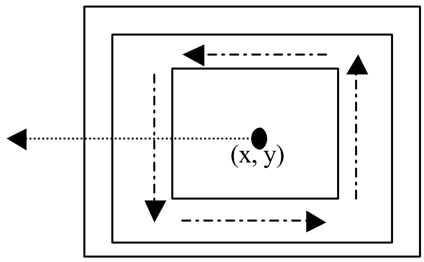


Figure 2. The expanding of searching area.

4. RESULTS AND COMPARISON

We implement the algorithm in Windows 2000 platform, with MS Visual C++ 6.0. Memory capacity of the computer is 128M and CPU is PII 266. Figure 3 consists of 4 frames of grayscale video clip of wave. The corresponding result frames, by Welsh's image processing extension are displayed in Figure 4. The corresponding results of our algorithm are presented in Figure 5. Figure 6 shows 4 frames of a popular grayscale cartoon video. The results of Welsh's extension lay out in Figure 7. We arrange our results in Figure 8. To compare them conveniently, Welsh's extension is adopted without swatch matching. Although, after swatch matching, perfect color image will be rendered in principle, it is a time-consuming process and requires a large quantity of interaction work. By comparison, we can easily conclude that the results of our algorithm are better than that of Welsh's extension.

Then, we can compare the processing time of Welsh's with ours.

Video clip	Welsh's extension	Our algorithm
Wave	108.666s	3.375s
Brain	111.941s	2.386s
Cartoon	106.724s	3.266s

Table 1. Processing Time of Welsh's and ours (per frame)

Here, the interaction time of Welsh's is not included, which is depended on the user. The processing time per frame of our algorithm varies with different images, the mean time is listed in Table 1. Conclusion will be easily drawn that our algorithm is much faster than Welsh's extension, as Welsh's algorithm mainly aims at image colorization instead of video colorization.

5. CONCLUSION

Although Welsh's image colorization algorithm can be naturally extended to process video clip, it is time-consuming and requires laborious interaction and skill of user, in order to eliminate abnormity in some swatches. This paper presents an algorithm of transferring color of key frames in a video clip to in-between frames, taking advantage of spatial and temporal correlation. Experimental results show color video of satisfactory quality can be rendered with quite fast processing speed.

6. ACKNOWLEDGEMENTS

This research work is co-supported by National Key Basic Research Project(973)(with project no.: 2002G3312100), NSF(60033010), Excellent Youth NSF of Zhejiang Province(RC 40008) and TRAPOYT Program in Higher Education Institutions of MOE, P.R.C.

7. REFERENCES

[1] Barron J.L., Fleet D.L. and Beauchemin S.S., *Performance of optical flow techniques*, International Journal of Computer Vision 12, 1, 43-77, 1994

[2] Black M.J. and Anandan P., The robust estimation of multiple motions: parametric and piecewise-smooth flow fields, Computer Vision and Image Understanding 63, 1, 75-104, 1996

[3] Chuang Y.Y., Agarwala A., Curless B., Salesin D.H. and Szeliski R., *Video Matting of Complex Scenes*, In Proceedings of Siggraph 2002, 243-248

[4] De Bonet J., *Multiresolution sampling procedure for analysis and synthesis of texture images*. In Proceedings of Siggraph 1997, 361-368

[5] Efros A.A. and Freeman W.T., *Image Quilting for Texture Synthesis and Transfer*, In Proceedings of Siggraph 2001, 341-346

[6] Gonzalez R.C. and Wintz P., *Digital Image Processing*, Addison-Wesley Publishin, Reading MA, 1987

[7] Hertzmann A., Jacobs C., Oliver N., Curless B. and Salesin D., *Iamge Analogies*, Proceedings of Siggraph 2001, 341-346

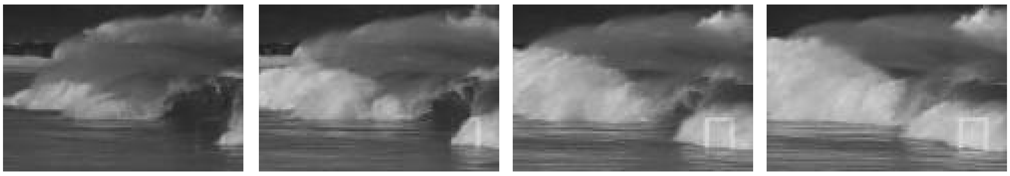
[8] Reinhard E., Ashikhmin M., Gooch B. and Shirley P., *Color Transfer between Images*, IEEE Computer Graphics and Applications, September/October 2001, 34-40

[9] Ruderman D.L., Cromin T.W. and Chiao C.C., *Statistics of Cone Responses to Natural Images: Implications for Visual Coding*, J.Optical Soc. Of America, 1998, vol 15, no.8, 2036-2045.

[10] Shi J.Y. and Pan Z.G., Multimedia Computer Architecture and Application, People's Postage Press, 1999

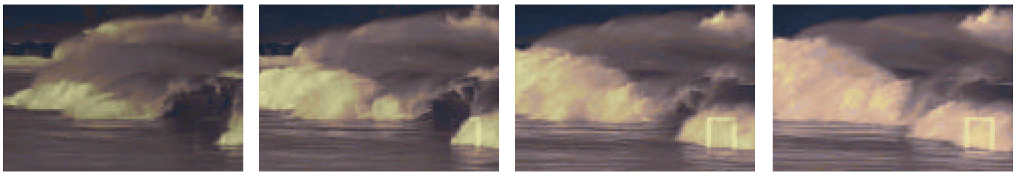
[11] Smith J.F. and Blinn J.F., *Blue Screen Matting*, In Proceedings of Siggraph 1996, 259-268

[12] Welsh T., Ashikhmin M. and Mueller K., *Transferring Color to Greyscale Images*, In Proceedings of Siggraph 2002, 277-280



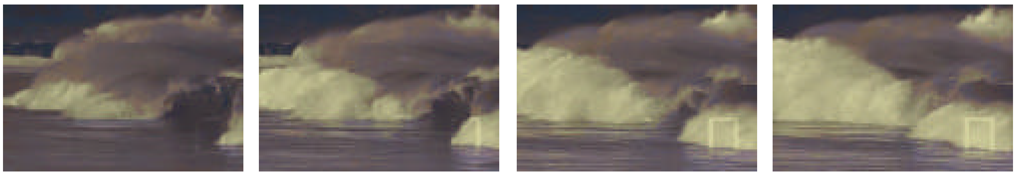
(a) 1st frame (b) 21st frame (c) 41st frame (d) 61st frame

Figure 3. Grayscale wave video



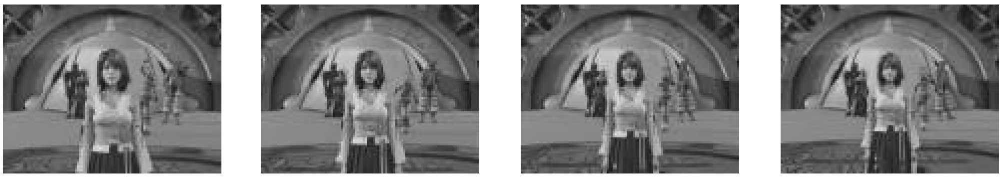
(a) 1st frame (b) 21st frame (c) 41st frame (d) 61st frame

Figure 4. Corresponding results of Figure 3. by Welsh's extension



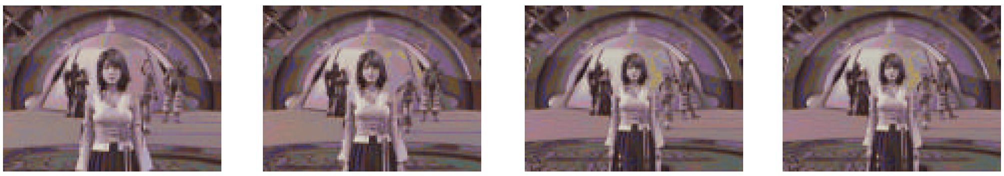
(a) 1st frame (b) 21st frame (c) 41st frame (d) 61st frame

Figure 5. Corresponding results of Figure 3. by our algorithm



(a) 1st frame (b) 21st frame (c) 41st frame (d) 61st frame

Figure 6. Grayscale cartoon video



(a) 1st frame (b) 21st frame (c) 41st frame (d) 61st frame

Figure 7. Corresponding results of Figure 6. by Welsh's extension

(a) 1st frame (b) 21st frame (c) 41st frame (d) 61st frame

Figure 8. Corresponding results of Figure 6. by our algorithm