Colorization Algorithm for Monochrome Image Sequences Using Optical Flow

Takahiko Horiuchi and Hiroaki Kotera
Department of Information and Image Sciences, Chiba University, Chiba, Japan

Abstract

Recently, the study of colorization algorithm for monochrome still images has prospered. Considering actual application, an extension to moving images becomes important. As the first step, this paper proposes two colorization algorithms for monochrome image sequences which are defined as continuous frames without scene changing. In our approach, key frames in the image sequence are colorized by a conventional colorization technique for still images or human interactive painting. Then, each color in the colorized frame propagates to the next monochrome frame by extracting displacement vector for each pixel between two frames. In this paper, the displacement vector is calculated by an optical flow algorithm, and color propagation to temporary direction is performed. Experimental results suggest a success of colorization for image sequences by the proposed techniques.

Introduction

Colorization is a computerized process that adds color to a black and white print, movie and TV program, supposedly invented by Wilson Markle. It was initially used in 1970 to add color to footage of the moon from the Apollo mission. The demand of adding color to monochrome images such as BW movies and BW photos has been increasing. For example, in the amusement field, many movies and video clips have been colorized by human's labor, and many monochrome images have been distributed as vivid color images. In other fields such as archaeology dealing with historical monochrome data and security dealing with monochrome images by a crime prevention camera, we can imagine easily that colorization techniques are useful.

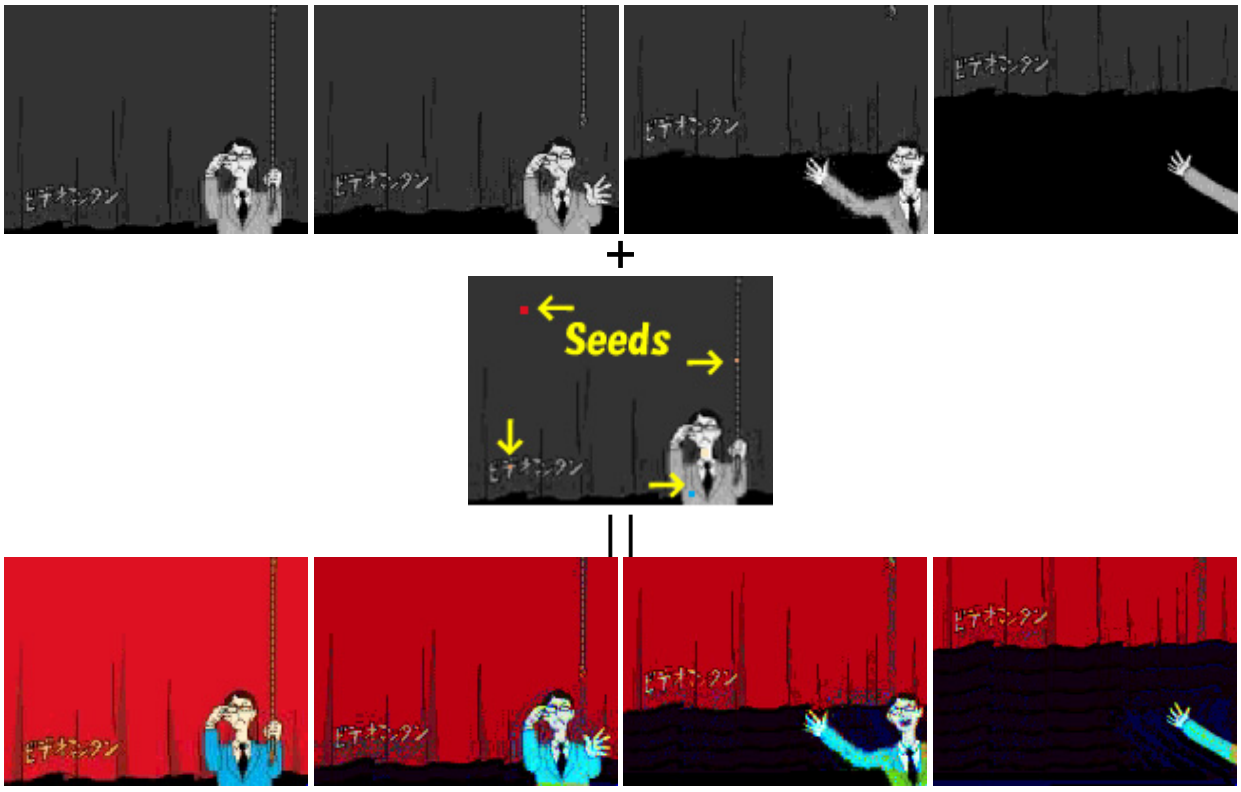


Figure 1. A sequence of monochrome images (upper) is colorized (lower) by giving a few color seeds to the first frame (middle).

A luminance value of a monochrome image can be assigned uniquely by a linear combination of an RGB color vector. However, searching for the RGB vector from a luminance value poses conversely an ill-posed problem, because there are several corresponding colors to one luminance value. Due to these ambiguous, human interaction usually plays a large role in the colorization process. The correspondence between a color and a luminance value is determined through common sense (green for grass, blue for the ocean) or by investigation. Even in the case of pseudo-coloring,[1] where the mapping of luminance values to an RGB vector is automatic, the choice of the color-map is purely subjective. Since there are a few industrial software products, those technical algorithms are generally not available. However, by operating those software products, it turns out that humans must meticulously hand-color each of the individual image subjectivity. There also exist a few patents for colorization.[2,3] However, those approaches depend on heavy human operation.

In 2002, Horiuchi et al. proposed a coloring method by sawing seed colors and propagating colors to the remaining pixels in the still image.[4-6] The technique works well by considering spatial continuity. However, the method was developed for still images and they cannot be applied to moving images.

In the same year, Welsh et al. proposed another coloring technique by transferring color from a reference color image.[7] The concept of transferring color from one image to another image was inspired by work in Ref. 8. A fast algorithm for Welsh's technique has also developed.[9] In the Welsh's technique, the user must prepare a reference image containing the desired colors with similar textures. Also, those methods do not explicitly enforce spatial continuity of the colors, and in some images it may assign vastly different colors to neighboring pixels that have similar intensities. Furthermore, the Welsh's technique does not use temporal information for colorization to sequential images.

This study aims to develop a colorization algorithm for monochrome moving images by considering with both space and temporal relations. As the first step, this paper proposes two colorization algorithms for monochrome image sequences, which are defined as continuous frames without scene changing.

Problem Setting

Consider an image sequence including N frames and a luminance value at (x,y) in n-th frame is expressed as $f^{(n)}(x,y):1\le n\le N$. Here, it is assumed that the image sequence must be composed of continuous scenes without scene changing. Scene changing problem may be solved by backward colorization. Each frame is $M_1\times M_2$ monochrome image. Let $\vec{I}^{(n)}(x,y)$ be a correct RGB color vector $(R(x,y),G(x,y),B(x,y))$ in n-th frame. Each element is quantized by L_1 bits. It is known that a color vector $\vec{I}^{(n)}(x,y)$ and the luminance value $f^{(n)}(x,y)$ have the following relation:

$$f^{(n)}=[0.299,0.587,0.114]\vec{I}^{(n)T} \tag{1}$$

where symbol T expresses transposition of a matrix. By Eq.(1), $f^{(n)}(x,y)$ can be given uniquely from $\vec{I}^{(n)}(x,y)$. Meanwhile, $\vec{I}^{(n)}(x,y)$ cannot be given uniquely from $f^{(n)}(x,y)$. Let $\{\vec{I}_i^{(n)}(x,y)\}$ be a set of *candidate colors* which satisfy Eq.(1) for a luminance value $f^{(n)}(x,y)$.

There are some key frames in the image sequence and the frames have already colorized by conventional colorization algorithm for still images or human's labor or any other techniques. So, the problem in this paper is how to propagate the color from the key frame to the next monochrome frame.

Colorization Algorithm for Image Sequences

The proposed algorithm is composed of two steps. The first step is displacement vector extraction and the second step is color estimation. Each step is described in the following subsections.

Displacement Vector Extraction by Optical Flow

In order to propagate colors in a colorized frame, displacement vector between adjacent frames is calculated for each pixel. In this paper, we use a typical optical flow algorithm. The optical flow of an image sequence is a set of vector fields, relating each frame to the next. Each vector field represents the apparent displacement of each pixel from frame to frame. If we assume the pixels conserve their luminance value, we arrive at the "brightness conservation equation",

$$f^{(n)}(x,y)=f^{(n-dn)}(x+dx,y+dy) \tag{2}$$

where (dx,dy) is the displacement vector for the pixel at coordinate (x,y) and n and dn are the frame and temporal displacement of the image sequence.

The obvious solution to Eq.(2) is to use template-based search strategies. A template of a certain size around each pixel is created and the best match is searched for in the next frame. The best match is usually found using correlation, sum of absolute difference or sum of squared difference metrics. In this paper, we use the following criterion :

$$\sum_p\sum_q\left|f^{(n)}(x+p,y+q)-f^{(n-1)}(x+dx+p,y+dy+q)\right| \tag{3}$$

where (p,q) shows an coordinate in template block. A vector (dx,dy) with the minimum Eq.(3) is determined as the displacement vector for $f^{(n)}(x,y)$. This process is often referred to as block-matching.[10] Such a search strategy is computationally costly and generally does not represent sub-pixel displacements.

Color Estimation

By the algorithm in previous subsection, a corresponding color pixel $\vec{I}^{*(n-1)}(x+dx,y+dy)$ in a colorized (n-1)-th frame is determined for each pixel

$f^{(n)}(x,y)$ in n-th monochrome frame. In this subsection, we propose two estimation algorithms.

(Algorithm 1)

$$\vec{I}^{*(n)}(x,y) = \vec{I}^{*(n-1)}(x+dx, y+dy) \tag{4}$$

Equation (4) means that each pixel (x,y) in n-th frame is colorized by the corresponding color $\vec{I}^{*(n-1)}(x+dx, y+dy)$ in (n-1)-th frame.

(Algorithm 2)

$\vec{I}^{*(n)}(x,y) = \vec{I}_i^{(n)}(x,y)$ which minimizes the difference between a *candidate color* $\vec{I}_i^{(n)}(x,y)$ at pixel (x,y) in n-th frame and $\vec{I}^{*(n-1)}(x+dx, y+dy)$ at corresponding pixel in the previous frame:

$$J(\vec{I}_i^{(n)}) = \left\| \vec{I}^{*(n-1)}(x+dx, y+dy) - \vec{I}_i^{(n)}(x,y) \right\| \to \min. \tag{5}$$

Equation (5) means that each pixel (x,y) in n-th frame is colorized by selecting the most suitable color $\vec{I}_i^{(n)}(x,y)$ from *candidate colors* $\{\vec{I}_i^{(n)}(x,y)\}$ for the luminance value $f^{(n)}(x,y)$. Here, the color $\vec{I}_i^{(n)}(x,y)$ has the minimum RGB difference between the corresponding color $\vec{I}^{*(n-1)}(x+dx, y+dy)$ in (n-1)-th frame and *candidate colors* $\{\vec{I}_i^{(n)}(x,y)\}$. Algorithm 2 has a property to keep the luminance value for the colorized image.

Experiments

In order to verify the proposed algorithms, we carried out colorization for monochrome image sequences. We have collected two kinds of images which are animated images and natural images. All original images have color and they are converted into monochrome images by using Eq. (1). Only the first frame is given as original color and other monochrome frames are colorized in order by the proposed methods. We set the template block size in Eq. (3) as 7x7 empirically.

Colorization for Animated Images

Since animated images have little change of the color between frames, it seems that displacement vectors can be detected stably. Figure 1(upper) shows a partial monochrome image. Frame size is 160x120 and frame rate is 16[frames/sec]. We used 48 frames for test. For reference, the original color frames are shown in Fig. 2. Figure 3 shows the colorized results by Algorithm 1. Algorithm 1 kept color. But, the structure was broken by accumulated error of optical flow detection. Meanwhile, in Algorithm 2, although there were some mis-colorized parts, better results were obtained as shown in Fig. 1 (lower). Since the detected displacement vectors are the same, the difference depends on the proposed color estimation algorithm.

Figure 2. Original color frames (Frame #1, #25, #30 and #35).

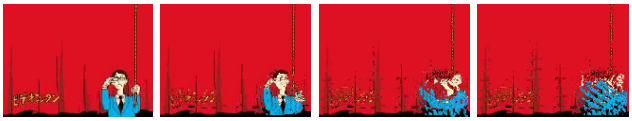


Figure 3. Colorized results by Algorithm 1.

In order to verify the results objectively, PSNR graph between an original color image and the colorized one is shown in Fig. 4. The definition of PSNR is as follows:

$$\begin{aligned}(PSNR) &= 10\log_{10}\frac{(2^{L_1}-1)^2}{E.M.S.} \\ &= 10\log_{10}\frac{3M_1M_2(2^{L_1}-1)^2}{\sum_{x,y}|I-I^*|^2}\end{aligned} \tag{6}$$

As shown in Fig.4, Algorithm 1 works well until 25th frames. From the 25th frames, red curtain goes up. Since the black part is increasing, PSNR of Algorithm 2 is going up from the 25th frame. The difference in coloring to a motion of curtain becomes the difference of the PSNR .

Colorization for Natural Scenes

The proposed algorithms were applied also to the several natural scenes. The performance of the proposed algorithms is verified using the image sequence in Fig. 5(a). Frame size is 320x240 and frame rate is 12[frames/sec]. We used 60 frames for test. Figures 5(b) and (c) show the colorized results. In the case of Algorithm 1, the preservability of a color was recognized as the same as the experimental results. But, the structure was also broken. On the other hand, in the case of Algorithm2, the preservability of a structure was accepted. However, faded color appeared more remarkably than colorized results for animated images.

PSNR graph is shown in Fig. 6. Although both PSNR were decreased gently, the slope of Algorithm2 is more sudden. The same results were obtained to other test images. These results are in agreement with subjective evaluation.

In the proposed methods, only the first frame was colorized as key frame beforehand. In order to avoid the influence of an accumulation error, we should increase the number of key frames. We also conducted the experiment which set up the key frame at the fixed interval. However, boundary frames were recognized, and it became colorized frames with sense of incongruity. Therefore, the coloring method using the frame of order such as MPEG algorithm will be required in order to carry out higher accurate colorization.

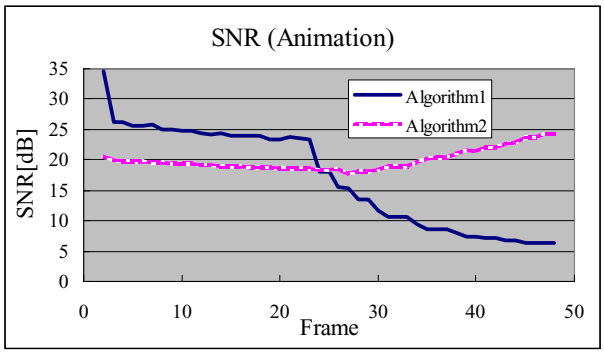
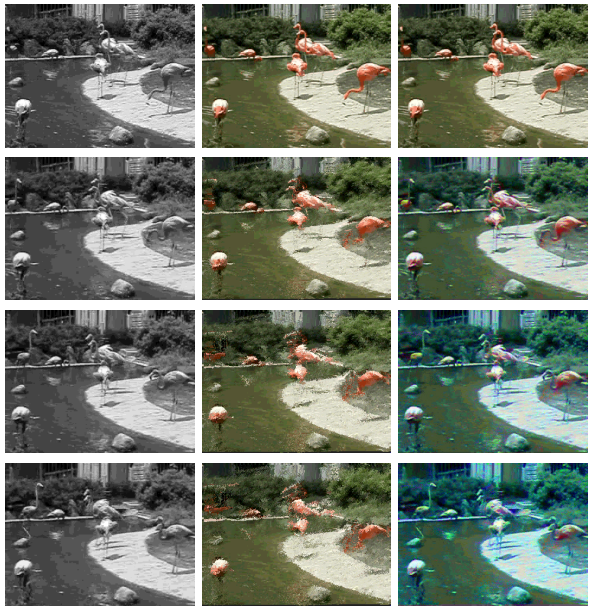


Figure 4. PSNR for each frame (Animation).



(a) Input frames (b)Algorithm 1 (c) Algorithm 2

Figure 5. Monochrome and colorized images "Birds" (Frame #1, #10, #20 and #30).

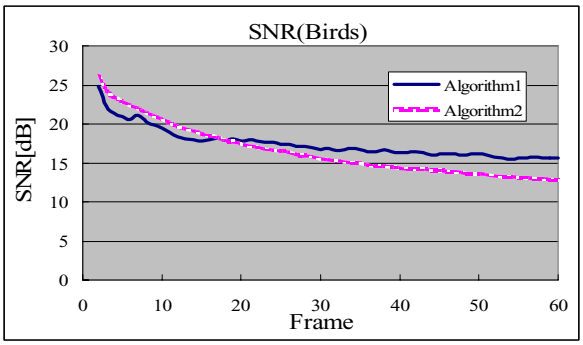


Figure 6. PSNR for each frame (Birds).

Conclusion

This paper proposed algorithms for colorizing monochrome image sequences. In our algorithms, the first frame is colorized by conventional algorithms for still images, and colors are propagated to remaining frames by optical flow. For color estimation, we proposed two kinds of algorithms. Experimental results showed that one algorithm can keep color and another algorithm can keep structure of scenes.

Error of displacement vector detection accumulates from frame to frame. So, we must develop more accurate displacement vector detection algorithm. Moreover, it may be necessary to use the information on a frame bi-directionally to the temporary axis.

References

1. R. C. Gonzales and P. Wintz, Digital Image Processing, (Addison-Wesley Publishing 1987).
2. M. Wilson and H. Brian, Coloring a black and white signal using motion detection, Canadian Patent, No.CA 01291260, 1991.
3. P. V. Roy, Designing, drawing, and colorizing generated images by computer, U.S. Patent, No.5,831,633, 1998.
4. T. Horiuchi, Estimation of Color for Gray-level Image by Probabilistic Relaxation, Proc. IEEE International Conference on Pattern Recognition, vol.3, pp.867-870, 2002.
5. T. Horiuchi and S. Hirano, Colorization algorithm for grayscale image by propagating seed pixels, Proc. IEEE International Conference on Image Processing, 2003.
6. T. Horiuchi, Colorization algorithm using probabilistic relaxation, Image and Vision Computing, vol.22, no.3, pp.197-202, 2004.
7. T. Welsh, M. Ashikhmin and K. Mueller, Transferring color to grayscale image, Proc. ACM SIGGRAPH 2002, vol.20, no.3, pp.277-280, 2002.
8. E. Reinhard, M. Ashikhmin, B.Gooch and P.Shirley, Color transfer between images, IEEE Computer Graphics and Applications, Sep./Oct., pp.34-40, 2001.
9. G. D. Blasi and D.-R. Recupero, Fast Colorization of Gray Images, Proc. Eurographics Italian Chapter 2003.
10. J. Little and A.Verri, "Analysis of Differential and Matching Methods for Optical Flow," Proc. IEEE Workshop on Visual Motion, pp.173-180, 1989.

Biography

Takahiko Horiuchi received his B.E., M.E. and Ph.D. degrees from University of Tsukuba in 1990, 1993 and 1995, respectively. He was a member of the Promotion of Science for Japanese Junior Scientists from 1993 to 1995. From 1995 to 1998, he was an Assistant Professor with the Institute of Information Sciences and Electronics, University of Tsukuba. From 1998 to 2003, he was an Associate Professor with the Faculty of Software and Information Sciences, Iwate Prefectural University. In 2003, he moved to Chiba University. He is an Associate Professor at Department of Information and Image Sciences.