Colorization for Monochrome Image with Texture

Takahiko Horiuchi and Hiroaki Kotera
Faculty of Engineering, Chiba University, Chiba, Japan

Abstract

Colorization is a computerized process of adding color to a monochrome image. The authors have developed colorization algorithms which propagate colors from seeded color pixels. Since those algorithms are constructed based on a region growing approach, failure colorization occurs at the place where a luminance changes intensely such as edge and texture. Although we developed, in the previous work, a partitioning algorithm for preventing the error propagation at edge, numerous color seeds were required for accurate colorization of the image with texture. This paper presents a new algorithm for colorization with texture by blending seeded colors. In our algorithm, the color can be estimated depending on the Euclidean distance and the luminance distance between each pixel to be colorized. It is shown that the proposed approach can be successfully applied to the images with texture by sowing a small number of color seeds.

Introduction

Colorization is a computerized process that adds color to a black and white print, movie and TV pro-gram, 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 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.

A luminance value of a monochrome image can be calculated uniquely by a linear combination of RGB color components. However, searching for the RGB components from a luminance value poses conversely an ill-posed problem, because there is several corresponding color 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 a set of RGB components 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.

Recently, simple colorization algorithms have been proposed by a few research groups[4-12]. In 2002, one of the authors proposed a colorization algorithm in which a small number of color seeds were sown on a monochrome image and the remaining pixels are colorized by propagating seeds' color to adjacent pixels[4]. The algorithm has been improved in Refs.5-9. In the same year, Welsh et al. colorize a monochrome image by transferring color from a reference color image with a stochastic matching[10]. The concept of transferring color from one image to another image was inspired by work in Ref.11. In the Welsh's method, the source image, which is the same kind of image as a monochrome image, is prepared and the colorization is performed by color matching between both pictures. After that, Levin et al. mark a monochrome image with some color scribbles and adjacent pixels are colorized by formulating and solving an optimization problem[12].

Those conventional algorithms are very simple and work well as an intuitive impression, especially for the image which can be segmented to a few large regions with the same chrominance components. However, it was difficult to perform accurate colorization for texture. In order to obtain an accurate colorized result for a texture image, Horiuchi's algorithm requires many color seeds. In the case of Welsh's algorithm, a specific reference image is required and Levin's algorithm requires many color scribbles.

This study aims to develop a new colorization algorithm for monochrome images with texture. This paper organized as follows: Section 2 presents our conventional algorithm and shows the problem for an image with texture. Section 3 presents the proposed colorization algorithm, and Section 4 demonstrates experiments. Finally, we conclude with a discussion in Section 5.

Conventional Colorization by Propagating Seeded Colors Independently

The most advanced colorization algorithm for still monochrome images by the authors' works can be shown in Ref.7. In this section, the conventional algorithm is explained briefly and a problem about texture is shown.

Let $I=(x,y)$ be a pixel in an input monochrome image and let $\{S_p=(x_p,y_p)\}_{p=1}^{P}$ be a set of color seeds, where P is the total number of the seeds. The color seeds, which are color pixels strictly, are given manually as a prior knowledge by a user. The position of the seeds and their color are determined by the user. Note that the color must be chosen with keeping the luminance of the original monochrome pixels. We present our method in CIELAB color space, each monochrome pixel I is transformed into the luminance signal $L(I)$. Each color seeds S_p is also transformed into $L(S_p), a(S_p), b(S_p)$, respectively.

In Ref.7, each pixel I is colorized by $L(I), a(f(I)), b(f(I))$ in CIELAB color space. The function $f(\cdot)$ selects a color seeds which have the minimum Euclidean distance, and defined as:

$$f(I) = \left\{ S_p \middle| \min_p \| I - S_p \|^2 \right\} \quad (1)$$

where $\|\cdot\|^2$ means the Euclidean distance in the X-Y image space.

Figure 1 shows an example of colorization by using the algorithm in Ref.7. Figure 1(a) shows an input monochrome image and the position of color seeds expressed by red circles. Each seeds were sown at the center of the circle. In this example, seven seeds were sown on the monochrome image by the user. Figure 1(b) shows the colorized results. Better result was obtained.

Figure 2 shows another example. The image consists of texture such as petals and trees. By sowing five color seeds as shown in Fig.2(a), a failure colorized result was obtained as shown in Fig.2(b). In order to obtain more accurate result, the user has to sow color seeds for each small region in the texture. In actual application, it is impossible to sow numerous seeds on each region. Reference 7 also proposed a partitioning algorithm to prevent the error propagation at edge. However, it is difficult to determine a threshold of partition. Even if the user can set the partition, failure estimation will be occurred after collapsing the partition. Moreover, the method produces visible artifacts of block distortion.

In order to solve the problem, we propose a new colorization algorithm by blending seed color in the next section.

Proposed Colorization by Blending Many Seed Color

Decision of the Chrominance Components

In our algorithm, we use two properties of natural images. The first property is that pixels with similar luminance values should have similar colors. This property was used for solving colorization problem in Levin's algorithm[12]. The second property is that near pixels should have similar colors. This property was used in Horiuchi's algorithm[4]. In the proposed method, we express those

(a) Position of seven color seeds on the monochrome image.

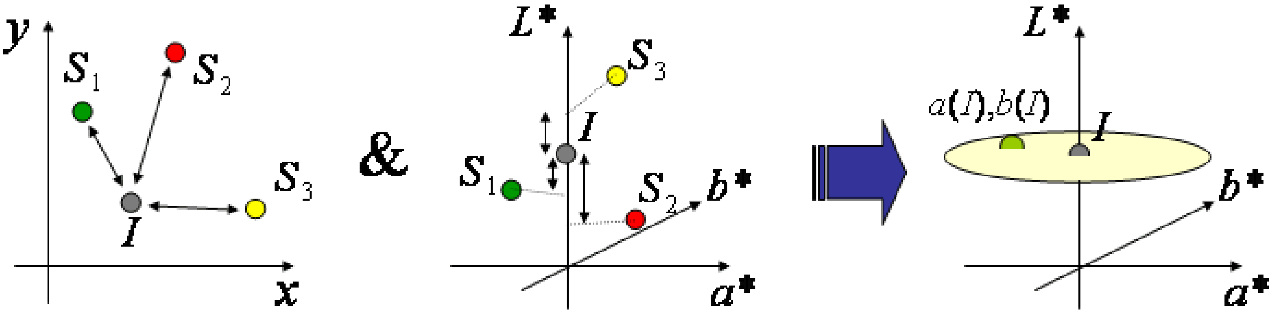
(b) Colorized image.

Figure 1. A colorized result by the method in Ref.7.

(a) Position of five color seeds on the monochrome image.

(b) Colorized image.

Figure 2. A failure example of colorization by the method in Ref.7.



*(a) **NED** on x-y plane.* *(b) **NLD** in CIELAB space.* *(c) Estimated chroma on a*-b* plane.*

Figure 3. Geometric expression of the proposed colorization process

two properties by distances ***NED*** and ***NLD*** as follows.

(NED: Normalized Euclidean distance)

We define the first distance $d_1(I,S_p) \in [0,1]$ between I and S_p by:

$$d_1(I,S_p) := \frac{\|I,S_p\|^2}{\sqrt{width^2 + height^2}} \tag{2}$$

where $width$ and $height$ means the horizontal size and vertical size of the monochrome image, respectively.

(NLD: Normalized luminance distance)

We define the second distance $d_2(I,S_p) \in [0,1]$ between I and S_p by:

$$d_2(I,S_p) := \frac{|L(I) - L(S_p)|}{100} \tag{3}$$

Then a contribution distance $d(I,S_p) \in [0,1]$ between I and S_p is defined by the weighted sum of the both distances as follows:

$$d(I,S_p) := \alpha d_1(I,S_p)^r + (1-\alpha) d_2(I,S_p)^r \tag{4}$$

where, $\alpha \in [0,1]$ means the constant value for weighting distances. In usually, $\alpha = 0.5$ is appropriately which will be described in Sec.4. In the case $\alpha = 1$, we can obtain almost the same colorized result as Ref.7. Symbol $r \in \Re^+$ means a factor which can change the influence of the color seeds, typically $r = 5$ is better in our experiments.

Equation (4) is translated into a weight for blending chrominance of color seeds $W(I,S_p) \in \Re^+$ as follows:

$$W(I,S_p) := \frac{d(I,S_p)^{-1}}{\sum_p d(I,S_p)^{-1}} \tag{5}$$

Then, the color for the pixel I can be represented by $L(I), a(I)$ and $b(I)$ as follows:

$$\begin{cases} L(I) \\ a(I) := \sum_p W(I,S_p) a(S_p) \\ b(I) := \sum_p W(I,S_p) b(S_p) \end{cases} \tag{6}$$

Figure 3 summarizes our algorithm. The proposed method uses two kinds of distances as shown in Fig.3(a) and Fig.3(b). Then a set of chrominance components (a^*,b^*) can be determined on the original luminance plane.

Gamut Mapping

The proposed method described in the previous subsection determines the chrominance components a^* and b^* by calculating the weighted average of them for color seeds in Eq.(4). Then the set of $L(I), a(I)$ and $b(I)$ is transferred into sRGB components. Sometimes, the determined color may exist outside of the sRGB gamut. Since we will see a colorized image as a printed image or displayed image, the estimated color must exist within sRGB gamut.

Several gamut mapping algorithms have been proposed in the field of color science. They are mostly designed to work in 2D L-

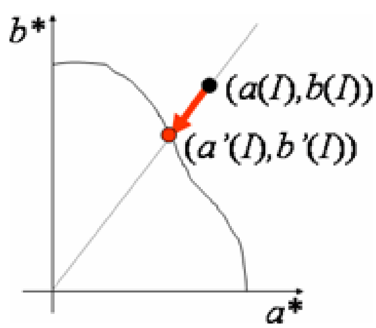


Figure 4. Gamut mapping on a-b* plane by clipping.*

(a) α=0.5.

(b) α=0.

Figure 5. Colorized results (r=5) for Fig.2(a) using the proposed method.

(a) r=1.

(b) r=9.

Figure6. Colorized results (α=0.5) for Fig.2(a) using the proposed method.

C(Lightness-Chroma) planes. However, the big assumption in the general colorization scheme is to keep the luminance value L^* of the original monochrome image. So, we consider a gamut mapping on a^*-b^* plane in this paper.

Figure 4 shows the overview of the gamut mapping used in our algorithm. We use a typical clipping algorithm in which a point out of gamut moves to the direction of origin. Therefore, the points within the gamut do not move to anywhere. In Fig.4, the estimated chrominance components $a(I)$ and $b(I)$ in Eq.(6) move to the boundary of the sRGB gamut $a'(I)$ and $b'(I)$, respectively.

Experiments

In order to verify the proposed algorithm, we carried out colorization for monochrome image with texture. Figures 5 and 6 show colorized results for Fig.2(a) by changing the parameters α and r in Eq.(4). The position and color of seeds were the same as Fig,2(a).

In Fig.5, the parameter r is constant ($r = 5$). In the case $\alpha = 0.5$, that is the same weight for both distances, beautiful color image was obtained as shown in Fig.5(a). In the case $\alpha = 0$, only the ***NLD*** is considered. So, interesting result was obtained as shown in Fig.5(b). Since a part of the flower was colorized by the green seed color of the ground, the cherry tree was expressed like a cherry tree with early leaves. In the case $\alpha = 1$, only the ***NED*** is considered. So, the algorithm is almost the same as the conventional method in Ref.[7] and the similar colorized image was obtained as Fig.2(b)..

Figure 6 shows a colorized result when the parameter α is constant ($\alpha = 0.5$). In the case $r = 1$, the weighted sum distance in Eq.(4) becomes linear. So, seed colors were strongly blended. The whole image became a sepia color. In the case $r = 9$, almost the same colorized image as Fig.5(a) ($r = 5$) was obtained. However, some vividness deteriorated.

Figure 7 showcases other examples of colorized images by the proposed method. Figures 7(a)-(c) demonstrate the use of the proposed algorithm with different types of images with textures. We set the constant parameters to be $\alpha = 0.5$ and $r = 5$ for all examples in Fig.7. The proposed technique works well on those

(a) Cat (512x384: 4 seeds, 22.7dB) (b) Trees (512x384: 8 seeds, 16.5dB) (c) Tower (480x640: 7 seeds, 22.3dB)

Figure 7. Examples of colorized images by the proposed method. (Upper) Monochrome input images. Color seeds were sown at the center of the red circles. (Lower) Colorized images.

images by sowing a small number of color seeds. In order to perform objective evaluation, we compared "original color" and "monochrome + synthesized color" by PSNR. Each PSNR is shown in the caption of Fig.7.

Figure 8 shows how the result is evolving with the addition of new seed by the user. At first, the user seeded a brown pixel on the neck. Then, all pixels were colorized from the brown seed by Eq.(6) as shown in Fig.8(a). Of course, all pixels were colorized by brown with keeping the original luminance. Next, the user seeded a white pixel on the chest, and all pixels were colorized from those two seeded pixels as shown in Fig.8(b). Then, third and forth seeds were sown on the background, and the user satisfied the colorized quality as shown in Figs.8(c) and (d).

In order to verify the relation between seeding position and colorized result, we sowed color seeds to different position from Fig.7(a). In the case of Fig.9(a), the brown seed was sown under the ear instead of neck, the brown hair on the chest was colorized by white which was sown on the chest. In this case, the user should sow other color seed one more on the chest to improve the result as shown in Fig.9(c). Then, a suitable colorized result can be obtained as shown in Fig.9(d). Of course, the proposed method is realized by an interactive software, the user can correct errors and to add or move color seeds as needed.

Conclusion

This paper proposed an algorithm for colorizing monochrome images with texture. In our algorithm, each color for monochrome pixels is estimated depending on two kinds of weighted distance which are Euclidean distance and luminance distance. Our assumption is that each monochrome pixel is more strongly affected in seed color with small distance. The proposed method does not need numerous color seeds and any region segmentations

(a) (b) (c) (d)

Figure 8. Colorization process. (a)First brown seed on the neck. (b) Second white seed on the chest. (c) Third green seed on the left background. (d) Forth green seed on the right background.

(a) Different seed position from Fig.7(a). (b) Colorized result from (a). (c) Adding one seed to (a). (d) Colorized result from (c).

Figure 9. The relation between colorized result and seeded position.

which were used in the previous works. We have shown the excellent colorization can be obtained for monochrome images with texture with a surprisingly small amount of user effort.

The proposed algorithm has two kinds of parameters α and r, typically $\alpha = 0.5$ and $r = 5$. As shown in Sec.4, interesting image can be obtained by changing those parameters. We have to investigate the effectiveness of those parameters as future subjects. Additionally, we plan to apply the proposed method to recoloring of color images.

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 ICPR2002, 3, pg. 867. (2002).
5. T. Horiuchi and S. Hirano, Colorization algorithm for grayscale image by propagating seed pixels, Proc. IEEE ICIP2003, pg.457. (2003).
6. T. Horiuchi, Colorization algorithm using probabil-istic relaxation, Image and Vision Computing, 22, 3, pg.197. (2004).
7. T. Takahama, T. Horiuchi and H.Kotera, Improvement on Colorization Accuracy by Partitioning Algorithm in CIELAB Color Space, Lecture Notes in Computer Science, Advances in Multimedia Information Processing – PCM2004, LNCS3332, PartII, (Springer, 2004) pg.794.
8. T. Horiuchi and H. Kotera, Colorization for Monochrome Image Based on Diffuse-Only Reflection Model, Proc. AIC 05, PartI, pg. 353. (2005).
9. M. Nishi, T. Horiuchi and H. Kotera, A Novel Picture Coding Using Colorization Technique, Proc. IS&T's NIP21, (2005).
10. T. Welsh, M. Ashikhmin and K. Mueller, Transfer-ring color to grayscale image, Proc. ACM SIGGRAPH 2002, 20, 3, pg.277. (2002).
11. E. Reinhard, M. Ashikhmin, B. Gooch and P. Shirley, Color transfer between images, IEEE Computer Graphics and Applications, Sep./Oct., pg.34. (2001).
12. A. Levin, D. Lischinski and Y. Weiss, Colorization using Optimization, ACM Trans. on Graphics, (2004).

Author 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 Science.

.