RELIABLE COLORIZATION ALGORITHM FOR IMAGES AND VIDEOS

Dae-Young Hyun†, *Jun-Hee Heu*†, *Chang-Su Kim*‡, *and Sang-Uk Lee*†

†School of Electrical Engineering and Computer Sciences,
Seoul National University, Seoul, Korea
‡School of Electrical Engineering, Korea University, Seoul, Korea
huanream@ipl.snu.ac.kr, junhee@snu.ac.kr, changsukim@korea.ac.kr, sanguk@ipl.snu.ac.kr

ABSTRACT

In this work, we propose an efficient image and video colorization algorithm. To achieve natural colorization, we first define initial color values and compute their reliabilities. For image colorization, initial colors are assigned by the user interaction. For video colorization, initial colors are transferred from the previous frame and their reliabilities are computed based on the motion information. Then, we formulate an energy function and minimize the function to refine the initial colors. Simulation results show that the proposed algorithm provides more natural colorization results on various images and videos than the conventional algorithms.

1. INTRODUCTION

Colorization is the process of adding colors to monochrome images or movies. Since the human visual system can perceive color information more efficiently than monochrome information, the value of monochrome images, films and TV programs can be increased through the colorization process. However, manual colorization consumes a lot of time and labor, especially in the case of a video sequence, consisting of a large number of still images. Therefore, it is essential to develop an efficient automatic or semi-automatic colorization algorithm.

Welsh *et al.* [1] proposed a colorization algorithm, which transfers colors from a source image to a gray target image by matching pixels based on luminance values and standard deviations. Semary *et al.* [2] improved the matching performance based on the texture classification. These color transferring algorithms provide stable results, provided that objects in input images have quite distinct luminance values and textures.

In another class of colorization algorithms, a user assigns colors to a selected set of pixels, which are then propagated to the whole image. Levin *et al.*'s algorithm [3] propagates colors, so that the difference between the color of a pixel and the weighted color average of the neighboring pixels is minimized. In their algorithm, the weight is inversely proportional to the luminance difference between a pixel and its neighbor. For video colorization, their algorithm employs temporal neighbors using an optical flow estimation scheme. Yatziv and Sapiro's algorithm [4] blends source colors to paint a target pixel based on the geodesic distances. A geodesic distance measures the variation of luminance values along a path between two pixels. For video colorization, they simply extended the geodesic distance to the 3-D case by considering temporally adjacent pixels. In general, these propagation algorithms [3, 4] provide better colorization results on still images than the color transferring algorithms [1, 2]. However, these propagation algorithms may yield color blurring artifacts, which can be propagated and amplified in video colorization.

Irony *et al.* [5] combined the notion of color transferring with the Levin *et al.*'s algorithm [3]. First, they matched pixels between a color image and a gray image based on the segmentation and learning method. Then, they used reliably matched pixels as color sources and colorized the other pixels using the Levin *et al.*'s algorithm. However, the Irony *et al.*'s algorithm inherits the drawback of the Levin *et al.*'s image colorization, although it improves the performance by transferring the color values of reliable pixels.

In this work, we propose a new colorization algorithm for images and videos. With a few brush strokes, a user first assigns color values to a few pixels of an image or the first frame of a video as initial color sources. For video colorization, except for the first frame, initial colors are temporally copied from the previous frame using the motion information. Then, the proposed algorithm computes the reliability of each initial color and uses the reliability information to formulate an energy function. Then, it minimizes the energy function to obtain the final colorization result. Simulation results demonstrate that the proposed algorithm provides much better and reliable colorization results than the conventional algorithms [3, 4].

The paper is organized as follows. The proposed colorization algorithm is described in Section 2. Experimental results are presented in Section 3. Finally, we discuss our approach and draw conclusions in Section 4.

2. PROPOSED ALGORITHM

Given the luminance information of an image or a video, the proposed algorithm uses an energy function with color reliability to reconstruct natural and realistic color values. We work in the YC_bC_r color space, where the Y is the luminance channel and C_b and C_r are the chrominance channels.

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea under the ITRC (Information Technology Research Center) Support program supervised by the IITA (Institute for Information Technology Advancement) (IITA-2009-C1090-0902-0018)

(a)

(b)

Figure 1: The colorization of the "Bird" image: (a) the input gray image with assigned color values and (b) the colorization result.

2.1 Initial Colors

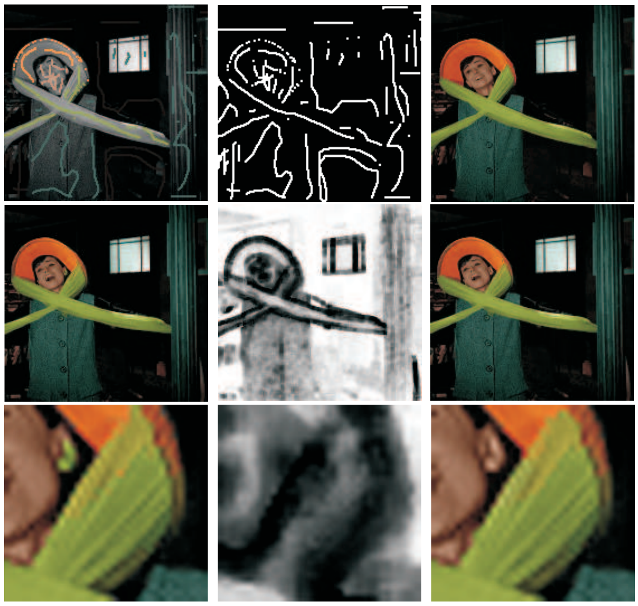
To colorize an image or the first frame of a video sequence, a user first scribbles colors on a selected set of pixels, as shown in Figure 1 (a) or the top image of Figure 2 (a). Let $C(\mathbf{p})$ denote the chrominance value C_b or C_r of pixel $\mathbf{p}$. The initial color value $\tilde{C}(\mathbf{p})$ is set to the assigned color if a user paints $\mathbf{p}$, or 128 otherwise.

In video colorization, video frames are colorized sequentially. Except for the first frame, the initial color of pixel $\mathbf{p}$ at the tth frame, $\tilde{C}_t(\mathbf{p})$, is temporally copied from $C_{t-1}(\mathbf{p}+\mathbf{m}_t(\mathbf{p}))$ using the motion vector $\mathbf{m}_t(\mathbf{p})$. The middle and bottom images of Figure 2 (a) show an example of this initialization. The motion vector is estimated to minimize the sum of squared differences (SSD) as follows.

$$\mathbf{m}_t(\mathbf{p}) = \arg\min_{\mathbf{m}\in\mathcal{S}} \mathrm{SSD}(\mathbf{m}), \tag{1}$$

$$\mathrm{SSD}(\mathbf{m}) = \sum_{\mathbf{x}\in\mathcal{B}(\mathbf{p})} |Y_t(\mathbf{x}) - Y_{t-1}(\mathbf{x}+\mathbf{m})|^2, \tag{2}$$

where $Y_t(\mathbf{x})$ denotes the luminance value of pixel $\mathbf{x}$ at the tth frame, $\mathcal{S}$ is a motion search range, and $\mathcal{B}(\mathbf{p})$ is a block centered at $\mathbf{p}$. In this work, the motion search range $\mathcal{S}$ is $[-7,7]\times[-7,7]$ and the block size of $\mathcal{B}(\mathbf{p})$ is 5×5.



(a) (b) (c)

Figure 2: From top to bottom, the colorization of the 27,892nd, the 27,905th frame and the enlarged part of the 27,905th frame in the "Funny Face" movie: (a) initial colors, (b) the reliabilities of the initial colors, and (c) final colorization results.

2.2 Reliabilities of Initial Colors

During the initial colorization, we also define the reliability $r(\mathbf{p})$ of each initial color $\tilde{C}(\mathbf{p})$. The reliability is a number within $[0,1]$: 1 for the most reliably colored pixel and 0 for the least reliably colored pixel. As shown in the top image of Figure 2 (b), for the first frame, $r(\mathbf{p})$ is set to 1 if a user paints pixel $\mathbf{p}$, and 0 otherwise.

Except for the first frame, the initial color $\tilde{C}_t(\mathbf{p})$ of the tth frame is transferred from $C_{t-1}(\mathbf{p}+\mathbf{m}_t(\mathbf{p}))$. Therefore, an incorrect motion vector $\mathbf{m}_t(\mathbf{p})$ may cause an incorrect initial color as shown in the middle and bottom images of Figure 2 (a). Moreover, errors in the previous frame may propagate to the current frame. Therefore, we define the reliability of $\tilde{C}_t(\mathbf{p})$ by

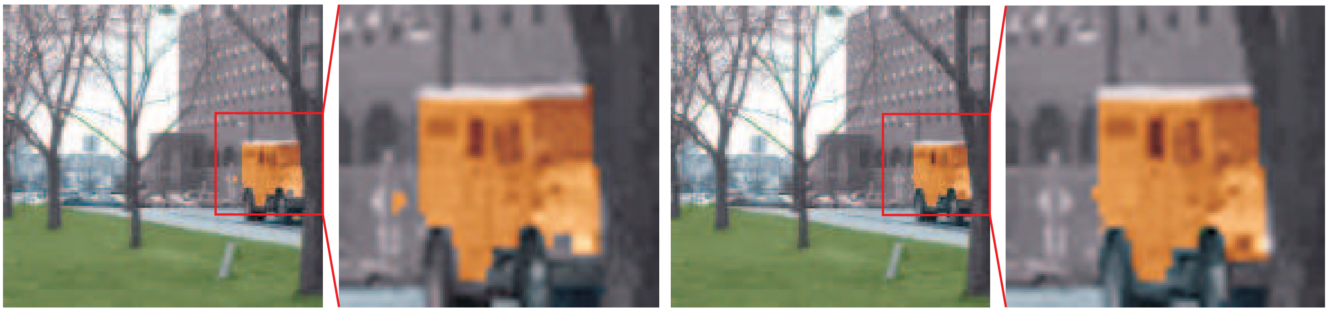
$$r_t(\mathbf{p}) = r_{t-1}(\mathbf{p}+\mathbf{m}_t(\mathbf{p}))\cdot \mathrm{MVR}_t(\mathbf{p}), \tag{3}$$

where $\mathrm{MVR}_t(\mathbf{p})$ stands for the motion vector reliability of $\mathbf{m}_t(\mathbf{p})$. We adopt the motion vector reliability in [6], given by

$$\mathrm{MVR}_t(\mathbf{p}) = \exp\left(-\beta(\min_{\mathbf{q}\in\mathcal{N}}\|\mathbf{m}_t(\mathbf{p})-\mathbf{m}_t(\mathbf{q})\|^2 + \mathrm{SSD}(\mathbf{m}_t(\mathbf{p}))\right), \tag{4}$$

where $\mathcal{N}$ is the set of neighboring pixels of $\mathbf{p}$. β is a positive scaling factor, and is fixed to 60 in this work. Thus, MVR is small if $\mathbf{p}$ has the motion vector that is too different from those of the neighboring pixels. Also, MVR is small if the motion vector is associated with a large SSD.

Note that the reliability $r_t(\mathbf{p})$ in Eq. (3) becomes lower, if the corresponding motion vector has a lower



(a) (b)

Figure 3: The colorization of the "Truck" video clip: (a) the 20th frame and its enlarged part and (b) the 21st frame and its enlarged part.

MVR or if the matching pixel $\mathbf{p} + \mathbf{m}_t(\mathbf{p})$ in the previous frame has a lower color reliability $r_{t-1}(\mathbf{p} + \mathbf{m}_t(\mathbf{p}))$. The middle and bottom images of Figure 2 (b) show the reliability map of the 27,905th frame, where a brighter pixel depicts a higher reliability. We see that low reliabilities are assigned to pixels near object boundaries, where initial colors tend to be unreliable.

2.3 Color Refinement

We define an energy function using the initial colors and their reliabilities. Then, we minimize the energy function to refine the colors and obtain the final colorization results. The energy function E consists of two terms, given by

$$E = \sum_{\mathbf{p} \in \mathcal{I}} E_1(C(\mathbf{p})) + \lambda \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{P}} E_2(C(\mathbf{p}), C(\mathbf{q})), \quad (5)$$

where $\mathcal{I}$ denotes the set of pixels in the image and $\mathcal{P}$ is the set of pairs of neighboring pixels in the image. The weighting coefficient λ is fixed to 5 in this work.

The first energy term $E_1(C(\mathbf{p}))$ measures a weighted difference between the initial color $\tilde{C}(\mathbf{p})$ and the desired color $C(\mathbf{p})$, which is given by

$$E_1(C(\mathbf{p})) = r(\mathbf{p}) \cdot \|C(\mathbf{p}) - \tilde{C}(\mathbf{p})\|^2, \quad (6)$$

where $r(\mathbf{p})$ is the reliability of the initial color $\tilde{C}(\mathbf{p})$. Thus, the first energy term indicates that a reliable initial color should not be changed drastically during the refinement. The second energy term $E_2(C(\mathbf{p}), C(\mathbf{q}))$ measures the color smoothness between neighboring pixels, given by

$$E_2(C(\mathbf{p}), C(\mathbf{q})) = W_{\mathbf{p},\mathbf{q}} \cdot \|C(\mathbf{p}) - C(\mathbf{q})\|^2, \quad (7)$$

where $W_{\mathbf{p},\mathbf{q}}$ is inversely proportional to the luminance difference

$$W_{\mathbf{p},\mathbf{q}} = \exp\left(\frac{-|Y(\mathbf{p}) - Y(\mathbf{q})|^2}{2 \cdot \sigma^2}\right) \quad (8)$$

and σ^2 is the variance of the luminance values in the image.

We minimize the energy function E in Eq. (5) using the graph cut algorithm [7]. Notice that the energy function is designed so that reliable initial colors are not changed much during the refinement, whereas unreliable initial colors are modified and affected by more reliable neighboring colors. As illustrated in the middle and bottom images of Figure 2 (c), the proposed algorithm suppresses the artifacts in the initial colorization by minimizing the energy function, yielding a natural color image.

Finally, after the color refinement, we update the color reliability of each pixel. For the first frame, $r(\mathbf{p})$ is updated to 1. For the other frames, it is updated via

$$r_t(\mathbf{p}) \Leftarrow \alpha r_t(\mathbf{p}) + (1 - \alpha), \quad (9)$$

where α is a constant between 0 and 1. In the extreme case $\alpha = 1$, we do not trust the color refinement, and the refined color has the same reliability as the initial color. In the other extreme case $\alpha = 0$, all refined colors are assigned the reliability 1 and it is assumed that the final colorization result is perfect without any errors. In this work, we set $\alpha = 0.7$ and increase the reliability after the refinement, so that the smallest reliability is at least 0.3 after the refinement.

Figure 3 shows the colorization results of two consecutive frames in the "Truck" video clip. In the 20th frame, there is a colorization error. But, it is observed that the error does not propagate to the next 21st frame. This is because, if a pixel color in the 21st frame is initialized with an erroneous color in the 20th frame, it is assigned a low reliability by Eq. (3), and thus the initial error disappears during the color refinement.

3. EXPERIMENTAL RESULTS

We compare the colorization performance of the proposed algorithm with those of the Levin *et al.*'s algorithm in [3] and the Yatziv and Sapiro's algorithm in [4] on various test images and videos.

Figure 4 compares the results on the "Cosmos" image. We assign color values to a few pixels in Figure 4 (a). In general, the Levin *et al.*'s algorithm achieves natural results by minimizing the color differences between neighboring pixels. However, in Figure 4 (b), their algorithm causes blurring in the sky region, which is far from any assigned colors. In Figure 4 (c), the Yatziv and Sapiro's algorithm yields erroneous color propagation near the edges of petals, which have similar luminance

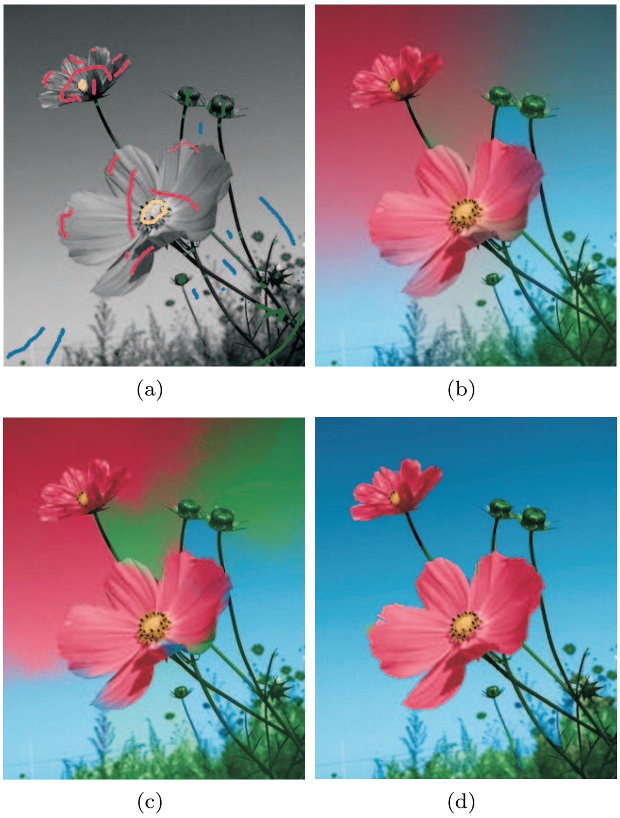


Figure 4: The colorization results on the "Cosmos" image: (a) the input monochrome image with assigned colors, (b) the Levin *et al.*'s algorithm, (c) the Yatziv and Sapiro's algorithm, and (d) the proposed algorithm.

to the background sky. On the contrary, the proposed algorithm provides much better quality reconstruction without any noticeable artifacts as shown in Figure 4 (d).

Figure 5 demonstrates that the proposed algorithm can be employed also to change the colors of a color image, *i.e.*, re-colorization. In Figure 5 (a), we assign new colors to the car body, and the white color to other parts. The white color indicates that the original color should be preserved as the initial color. In Figures 5 (b)$\sim$(d), we see that the car body is re-colorized realistically.

Figure 6 shows the video colorization results of the black and white film "Roman Holiday," which was made in 1953. In the first image of Figure 6 (a), a user first paints a few colors on the 108,515th frame, which is used as the first frame of the experiment. Using the those colors, we automatically colorize the subsequent frames. We see that the first frame is faithfully colorized by all the Levin *et al.*'s algorithm, the Yatziv and Sapiro's algorithm, and the proposed algorithm. However, we see that, in the subsequent frames, the Levin *et al.*'s algorithm causes severe blurring errors, especially on the woman's arm and man's face. Also, the Yatziv and Sapiro's algorithm yields errors on the man's face. On the other hand, the proposed algorithm provides better performance and colorizes the subsequent frames reliably without error propagation.

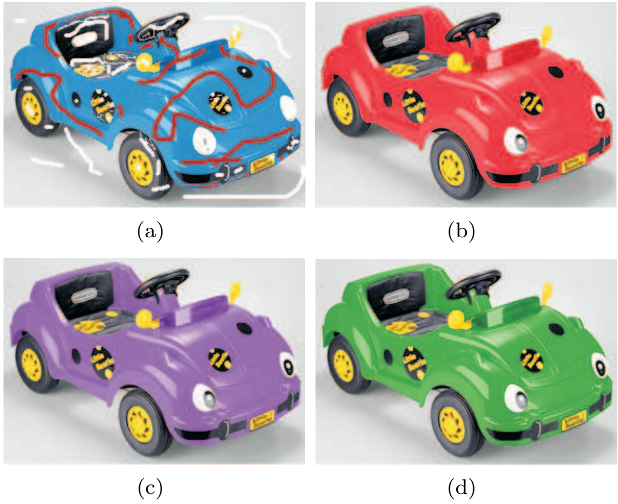


Figure 5: The re-colorization of the "Car" image: (a) the input color image with user scribbles, where the white color indicates that the original color should be preserved, and (b)$\sim$(d) the re-colorization results.

4. CONCLUSION

In this work, we proposed an efficient colorization algorithm for images and videos. The proposed algorithm first assigns initial colors and computes their reliabilities. Then, based on the reliability information, the proposed algorithm defines an energy function and minimizes the function to refine the colors. For the minimization, we employed the graph cut algorithm. Simulation results demonstrated that the proposed algorithm yields significantly better colorization results than the conventional algorithms.

REFERENCES

[1] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to grayscale images," in *Proc. ACM SIGGRAPH*, vol. 20, no. 3, July 2002, pp. 277–280.

[2] N. A. Semary, M. M. Hadhoud, W. S. E. Kilani, and N. A. Ismail, "Texture recognition based natural gray images coloring technique," in *Proc. National Radio Science Conference (NRSC)*, Mar. 2007, pp. 1–12.

[3] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 689–694, Aug. 2004.

[4] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Processing*, vol. 15, no. 5, pp. 1120–1129, May 2006.

[5] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Proc. Eurographics Symposium on Rendering*, June 2005, pp. 201–210.

[6] D. Wang, A. Vincent, and P. Blanchfield, "Hybrid de-interlacing algorithm based on motion vector reliability," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 1019–1025, Aug. 2006.

(a)

(b)

(c)

(d)

Figure 6: The colorization of the film "Roman Holiday": From left to right, the 108,515th, the 108,518th frame, an enlarged part of the 108,518th frame, and the 108,532nd frame. (a) the input monochrome frames, (b) the Levin *et al.*'s algorithm, (c) the Yatziv and Sapiro's algorithm, and (d) the proposed algorithm.

[7] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, Nov. 2006.