EDGE-PRESERVING COLORIZATION USING DATA-DRIVEN RANDOM WALKS WITH RESTART

Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee

Dept. of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea

thkim@diehard.snu.ac.kr, kyoungmu@snu.ac.kr, sanguk@ipl.snu.ac.kr

ABSTRACT

In this paper, we consider the colorization problem of grayscale images in which some color scribbles are initially given. Our proposed method is based on the weighted color blending of the scribbles. Unlike previous works which utilize the shortest distance as the blending weights, we employ a new intrinsic distance measure based on the Random Walks with Restart (RWR), known as a very successful technique for defining the relevance between two nodes in a graph. In our work, we devise new modified data-driven RWR framework that can incorporate locally adaptive and data-driven restarting probabilities. In this new framework, the restarting probability of each pixel becomes dependent on its edgeness, generated by the canny detector. Since this data-driven RWR enforces color consistency in the areas bounded by the edges, it produces more reliable edge-preserving colorization results that are less sensitive to the size and position of each scribble. Moreover, if the additional information about the scribbles which indicate the foreground object is available, our method can be readily applied to the object segmentation and matting. Experiments on several synthetic, cartoon and natural images demonstrate that our method achieves much high quality colorization results compared with the state-of-the-art methods.

Index Terms— Data-Driven Random Walks with Restart, color blending, edge-preserving colorization.

1. INTRODUCTION

Colorization problem is to naturally add colors to a grayscale image, in which some pixels, called scribbles, with initial user-defined colors are given. Recently, several colorization approaches [3][5] have been proposed. The method by Levin et al. [3] colorizes an image by minimizing a quadratic energy function derived from the color differences between a pixel and the weighted average of its neighborhood colors. Their colorization results are largely sensitive to the size and position of each scribble, and have over-smoothed color artifacts. The color blending method by Yatziv et al. [5] enables fast colorization by applying the shortest distance between a pixel and a scribble to the color blending weight. However, since the global color-continuity between neighboring pixels is not guaranteed, it consequently suffers from the weak boundary problem in finding weak boundaries when they are parts of a consistent boundary. Thus, a user is often left with the task of manually drawing the additional scribbles for delineating complicated boundaries between a subject and the background. It is an expensive and time-consuming process.

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

Fig. 1. Introducing our colorization framework. (a) Grayscale image with color scribbles. (b) Initial edge map, generated by Canny Edge Detector [1]. (c) Our colorization result, by refering the edge map (b). (d) Our matting result.

In this paper, we introduce a new colorization framework based on the concept of the weighted color blending, like [5]. Unlike the previous work [5], which uses the shortest distance for the blending weights, we propose to use the steady-state probability by a Random Walks with Restart (RWR). RWR, similar to the graph-based semi-supervised learning [6], is known as a very successful technique for defining the relevance between two nodes in a graphical model, such as in graph mining [4] and image segmentation [2]. In our work, we additionally utilize the edge map, generated by Canny Edge Detector [1], as shown in Fig. 1(b). By incorporating these edgeness values into the restarting probabilities of RWR, we design a new data-driven RWR to estimate the blending weights under the constraint whereby pixels in the areas bounded by the edges tend to have similar weights.

Our colorization method has various advantages over conventional approaches [3][5] as follows. First, since our data-driven RWR enforces color consistency in the edge-bounded regions, we can obtain more reliable edge-preserving colorization results that are less sensitive to the scribble properties: size and position (e.g. Fig. 1(c)). Second, the edge map guides the suitable positions of the scribbles to the user. Thus we can expect reasonable results according to the edge-guided user-inputs. Finally, if the additional information about scribbles which indicate the foreground object is available, our method can be readily applied to the object segmentation and matting problems (e.g. Fig. 1(d)).

2. PROPOSED ALGORITHM

We consider the colorization problem of adding colors to all N pixels $X = \{x_i\}_{i=1,...,N}$ in a grayscale image I in which some pixels, called K scribbles $S = \{S^1, ..., S^K\}$, with initial user-defined colors are given. We use YUV color space where $Y = \{y_i\}_{i=1,...,N}$ is the monochromatic luminance channel, which we will refer to it simply as intensity, while $U = \{u_i\}_{i=1,...,N}$ and $V = \{v_i\}_{i=1,...,N}$ are the chrominance channels that encode the color. Our goal is to complete both the U and V channels, given $Y = I$. We deal with the only U channel in this paper, since the V channel can be treated in the same manner.

Our colorization algorithm is based on the weighted color blending of the scribbles, like [5].

$$u_i = \frac{\sum_{k=1}^{K} \pi_i^k \tilde{u}_i^k}{\sum_{k=1}^{K} \pi_i^k}, \quad (1)$$

where $\tilde{u}_i^k$ is the estimated color of a pixel x_i from a scribble S^k, and π_i^k is the blending weight how much the color of a pixel x_i is affected by the color $\tilde{u}_i^k$, estimated from S^k. To solve (1), the blending weights $\Pi^k = \{\pi_i^k\}_{i=1,...,N}$ and colors $\tilde{\mathrm{U}}^k = \{\tilde{u}_i^k\}_{i=1,...,N}$ must be estimated from a scribble S^k. First, we focus on how to design the blending weights Π^k, derived from a distance function by the grayscale intensity difference. Since each blending weight π_i^k indicates the relevance score between a pixel x_i and a scribble S^k, we propose to use the steady-state probability that a particle starting at S^k stays at x_i in a data-driven way, by RWR [4][2]. Compared with the traditional shortest distance in [5], our relevance measure can capture the whole relationship between a pixel and a scribble according to the intensity geometry. Then, we design the cost function to estimate all pixel colors $\tilde{\mathrm{U}}^k$ from each scribble S^k. Let us explain about the computation of those two parts: blending weights Π^k and estimated colors $\tilde{\mathrm{U}}^k$, in detail.

2.1. Estimating Blending Weights

Given an image I, we first construct an undirected graph $G = (X, E)$, where each node $x_i \in X$ uniquely identifies an image pixel, and each edge $e_{ij} \in E$ spanning between the nodes $x_i, x_j \in X$ is determined by the small neighborhood system, which is usually chosen to be either 4 or 8 neighborhoods. Note that a intensity change causes a related change in color channel. Each weight $w_{ij} \in W$ is assigned to the edge e_{ij}, and defined as the typical Gaussian weighting function in the intensity channel Y as follows.

$$w_{ij} = exp\left(-\|y_i - y_j\|^2/2\sigma^2\right), \quad (2)$$

Algorithm 1 Outline of a data-driven RWR.

1: In a graph G, a particle $\mathcal{M}$ starts from any pixel of each scribble S^k with a probability $\frac{1}{|S^k|}$.

2: At each node x_i, return to **step 1** with a locally adaptive restarting probability c_i:

$$c_i = \begin{cases} \mu & \text{if } x_i \in S \text{ or } f(x_i) = 1 \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

where $0 < \mu < 1$, and $f(x_i)$ is an edge indicating function that provides the binary output with a value 1 if a pixel x_i is on the edge, and 0 otherwise.
Otherwise, $\mathcal{M}$ transmits to its neighborhood $v_j \in \aleph_i$ with the transition probability $(1 - c_i)p_{ij}$:

$$p_{ij} = \frac{w_{ij}}{\sum_{l=1}^{N} w_{lj}}, \quad (4)$$

where w_{ij} is defined in (2).

3: Until convergence, go to **step 2**.

where σ is the variance of the total pixel intensities. It provides us with a numerical measure for the color similarity between two neighboring pixels.

By calculating RWR scores between each pair of nodes in a data-driven way, we simply estimate the blending weights Π^k. The detailed process of our data-driven RWR is outlined in Algorithm 1. The higher value μ gets, the higher blending weights the seed pixels with user-given color have. Also, since it reduces the transition probability that a particle passes the edges, the color consistency in the edge-bounded regions is emphasized for the remaining pixels. Thus we can generate the edge-preserving colorization results.

Our data-driven RWR can be formulated in matrix form with a blending weight vector $\vec{\pi}^k = [\pi_i^k]_{N\times 1}$ as follows [2].

$$\begin{aligned} \vec{\pi}^k &= (\mathbf{I} - \mathbf{C})\mathbf{P}\vec{\pi}^k + \mathbf{C}\vec{b}^k \\ &= (\mathbf{I} - (\mathbf{I} - \mathbf{C})\mathbf{P})^{-1}\mathbf{C}\vec{b}^k, \end{aligned} \quad (5)$$

where $\mathbf{C} = diag(c_1, ..., c_N)$ and $\mathbf{P} = [p_{ij}]_{N\times N}^T$ are the restarting probability and transition matrices in (3) and (4), respectively. The vector $\vec{b}^k = [b_i^k]_{N\times 1}$ indicates the k-th scribble with $b_i^k = \frac{1}{|S^k|}$ if $x_i \in S^k$ and 0 otherwise. Thus, the the blending weights of all pixels can be easily obtained by solving (5) using a simple matrix inversion technique.

2.2. Estimating Pixel Colors

For estimating the blending colors $\tilde{\mathrm{U}}^k$ from S^k,we propose the following cost function $\mathcal{J}^k$ to minimize the color difference between each pair of pixels, as in [3].

$$\mathcal{J}^k = \sum_{x_i, x_j \in X} p_{ij}(\tilde{u}_i^k - \tilde{u}_j^k)^2 + \lambda \sum_{x_i \in S^k} (\tilde{u}_i^k - \hat{u}_i^k)^2, \quad (6)$$

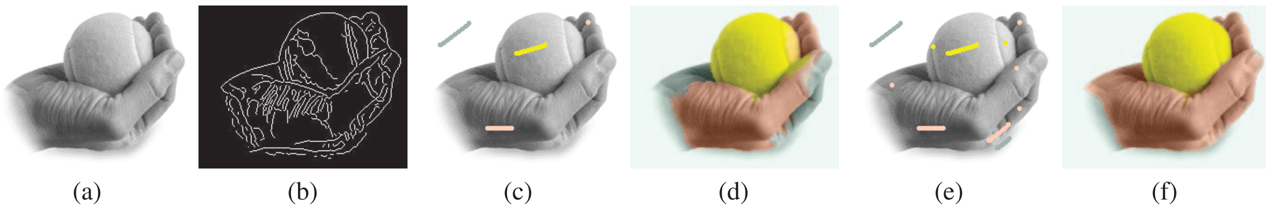

Fig. 2. An overview of our colorization algorithm. Given an input image (a), (c)-(f) are the examples of how the results are evolving with the addition of new scribbles by referring to the edge map in (b).

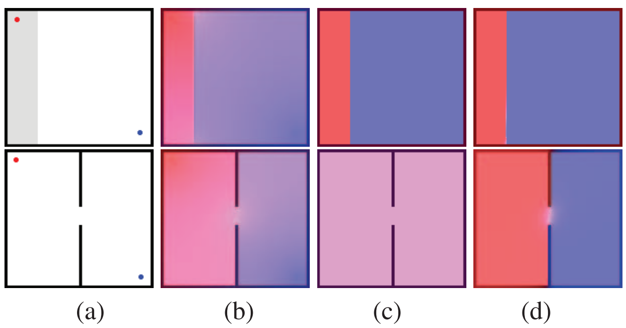

Fig. 3. The colorization tests with synthetic images containing a strong (complete) (top row) and weak (broken) boundary (bottom row). (a) Grayscale images with color scribbles. Colorizations by (b) Levin et al. [3]; (c) Yatziv et al. [5]; (d) Our algorithm.

where $\hat{\mathrm{U}}^k = \{\hat{u}_i^k\}_{1,\ldots,N}$ is the initial color vector, where $\hat{u}_i^k$ is assigned by the user-given color value if $x_i \in S^k$ and 0 otherwise. The first term of the right-hand side in (6) is the color-continuity constraint that two neighboring pixels should have similar colors if their intensities are similar. The second term is the unary constraint that the seed pixels in each scribble should maintain the user-given colors. For larger λ, each seed $x_i \in S^k$ gets more similar color to the user-given one $\hat{u}_i^k$. Thus λ represents the authenticity of the user-inputs. Since $\mathcal{J}^k$ is the quadratic equation, we simply minimize it by differentiating its matrix form with respect to the vector $[\tilde{u}_i^k]_{N\times 1}$, and set to zero. Note that unlike [5] where the colors are directly chosen among the scribble colors only, our estimated colors for each scribble consider the intensity variation as well.

3. EXPERIMENTS

Fig. 2 shows an overview of our proposed algorithm. Given a grayscale image in Fig. 2(a), we first compute the edge map, that can guide the suitable positions of the scribbles, in Fig. 2(b). If the colorization result with the initial scribbles (Fig. 2(c)) is unsatisfactory as in Fig. 2(d), a user can simply add some extra scribbles (Fig. 2(e)) and get more complete result as shown in Fig. 2(f).

Now, we evaluate our colorization algorithm using several synthetic, cartoon and natural images. Our method has two parameters: μ and λ in (3) and (6). In our experiments, these parameters were chosen empirically, and we set $\mu = 0.999$ and $\lambda = 10^3$ for all the test images. Fig. 3 shows the comparative results on two synthetic images with a strong and weak boundary, respectively. The results by Levin et al. [3] have blurred regions for both cases according to the intensity difference across the edge and the positions of the scribbles as shown in Fig. 3(b). In Fig. 3(c), the shortest-path-based method of Yatziv et al. [5] produced good colorization result regardless of the positions of the scribbles in the case of the clear boundary. However, it failed to detect the weak boundary. Compared with the two conventional methods, we observe that our algorithm produced successful results for both clear and weak boundary cases as depicted in Fig. 3(d).

Fig. 4 and Fig. 5 show the colorization results of our algorithm on several cartoon and natural images, respectively, compared with those of [3] and [5]. The cartoon and natural images often have many weak boundaries. Moreover, there can be thin elongated parts such as the cartoon outlines as in Fig. 4 and the dragonfly's leg in the middle row of Fig. 5. In these cases, the conventional methods [3][5] usually require for the users to delineate complicated boundaries between regions. However, by our approach, we can obtain much higher-quality results as in Fig. 4(e) and Fig. 5(e) by adding even less number of scribbles (or few pixels). Moreover, very accurate image matting results can be obtained from our blending weights and estimated colors, as shown in Fig. 4(f) and Fig. 5(f). These comparisons confirm the accuracy and robustness of our colorization algorithm.

4. CONCLUSIONS

In this paper, we propose a novel edge-preserving colorization framework by considering the color consistency in the regions bounded by the edges. This edge constraint is added into our colorization formulation by controlling the restarting probability of RWR according to the edgeness in a data-driven manner. Our work has several advantages. First, it produces high-quality colorization results in cartoon as well as natural images. This is because the global distance computed by RWR is designed as the blending weight, instead of the con-

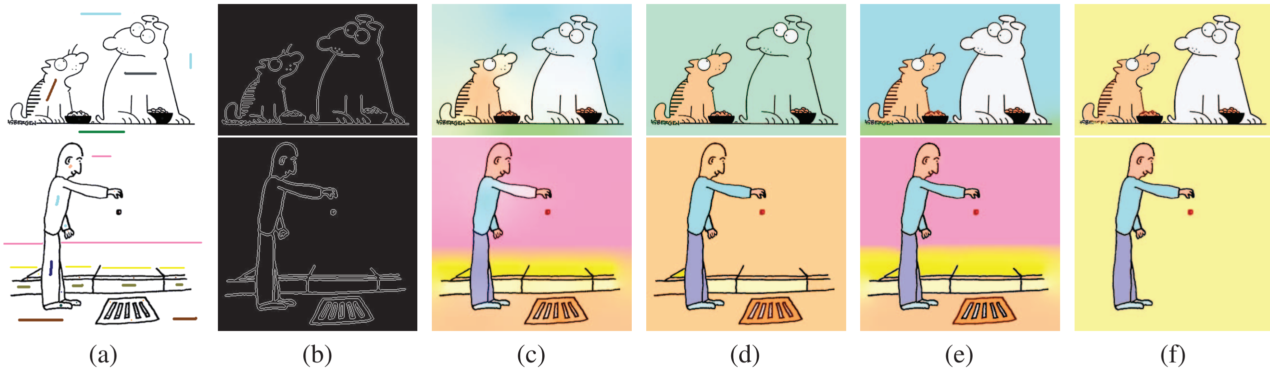

Fig. 4. The colorization tests on the cartoon images. (a) Input images with color scribbles. (b) Edge maps [1]. Colorizations by (c) Levin et al. [3]; (d) Yatziv et al. [5]; (e) Our algorithm. (f) Matting results by our algorithm.

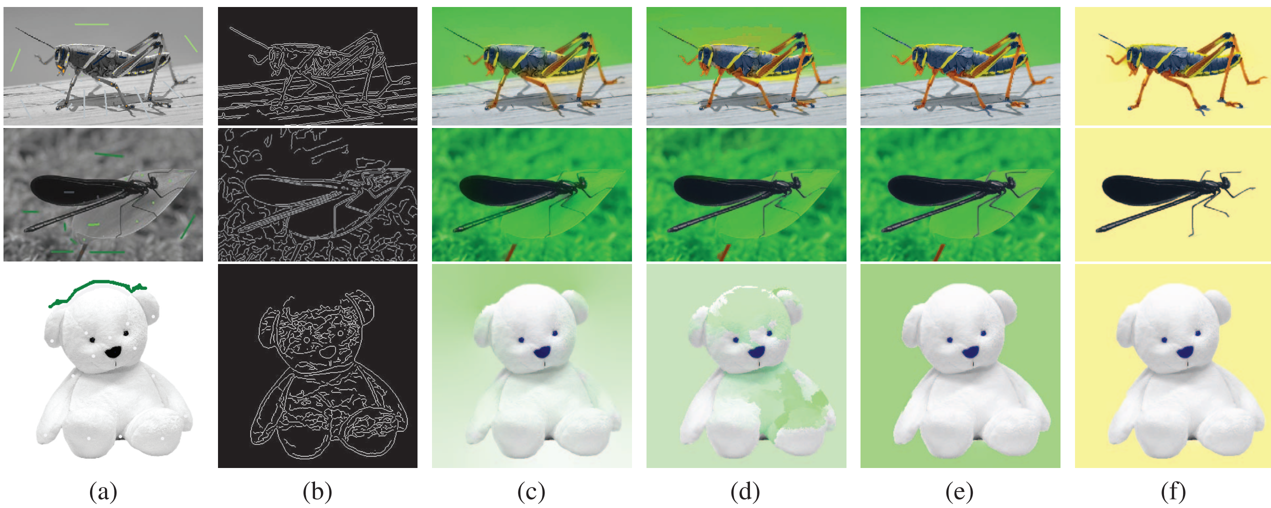

Fig. 5. The colorization tests on the natural images. (a) Input images with color scribbles. (b) Edge maps [1]. Colorizations by (c) Levin et al. [3]; (d) Yatziv et al. [5]; (e) Our algorithm. (f) Matting results by our algorithm.

ventional shortest distance. Moreover, it is easy to expect the resulted colorization by referring the edge map. Finally, our colorization results can be used for other applications such as image matting. Further research issues include the selection of the optimal parameter values. We are also interested in the combination of the coloization and matting frameworks.

Acknowledgement

This research was supported in part by the Defense Acquisition Program Administration and Agency for Defense Development, Korea, through IIRC (UD070007AD), and in part by the IT R&D program of MKE/IITA (2008-F-030-01, Development of Full 3D Reconstruction Technology for Broadcasting Communication Fusion).

5. REFERENCES

[1] J. F. Canny. A computational approach to edge detection. *PAMI*, 8(6):679–698, 1986.

[2] T. H. Kim, K. M. Lee, and S. U. Lee. Generative image segmentation using random walks with restart. In *ECCV*, 2008.

[3] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. 2004.

[4] J. Y. Pan, H. J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD*, 2004.

[5] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IP*, 15(5):1120–1129, 2006.

[6] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *NIPS*, 2003.