SELF-SIMILARITY-BASED IMAGE COLORIZATION

Jiahao Pang, Oscar C. Au*, Yukihiko Yamashita†, Yonggen Ling*, Yuanfang Guo*, Jin Zeng**

*The Hong Kong University of Science and Technology
†Tokyo Institute of Technology

ABSTRACT

In this work, we tackle the problem of coloring black-and-white images, which is image colorization. Existing image colorization algorithms can be categorized into two types: scribble-based colorization algorithms and example-based colorization algorithms. Differently, we propose a hybrid scheme that combines the advantages of both categories. Given the grayscale image to be colorized and a few color scribbles (or scattered color labels) as input, the proposed method manages to colorize the grayscale image with high quality. Similar to the mechanisms in example-based colorization methods, our algorithm firstly propagates chrominance information based on the assumption that similar image patches should have similar colors. Therefore colors of some pixels can be transferred from similar patches with known colors. After that, we apply scribble-based colorization algorithm to fully colorize the grayscale image, with different confidences assigned onto the transferred color labels. Experimental results show that, the proposed method effectively utilizes the known chrominance, and provides pleasant colorizations with very few user interventions.

***Index Terms*—** Colorization, self-similarity, optimization, image restoration, non-local method

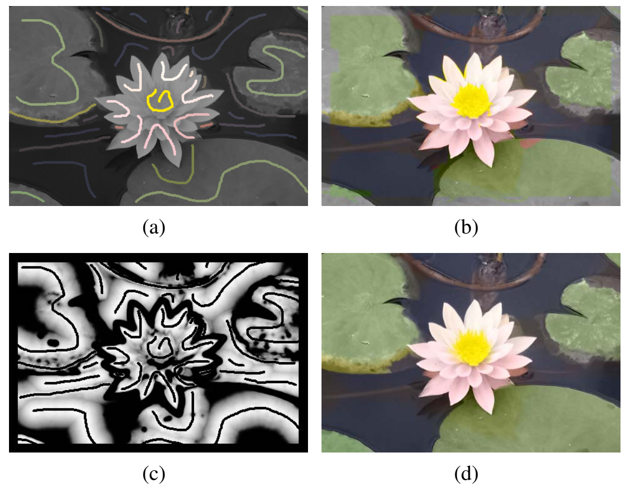


Fig. 1. (a) Grayscale image with marked color scribbles. (b) Propagated colors. (c) Confidence map of the propagated colors. (d) Colorization result of our method.

1. INTRODUCTION

Image colorization, which is the process of adding colors to a monochrome image, used to be a time-consuming and tedious task requiring tremendous user efforts. Although it is expensive, colorization is widely used for coloring black-and-white photos, image recoloring, color image compression, etc. To resolve this highly ill-posed problem, all existing methods require some color cues. According to different forms of given cues, image colorization algorithms can be categorized into two types: scribble-based colorization algorithms and example-based colorization algorithms.

In scribble-based colorization algorithms [1, 2, 3, 4], the user or another automatic method specifies the colors of certain regions of the image in form of either color scribbles or scattered color labels. Then the algorithms colorize the given grayscale image based on the known chrominance information. The seminal work by Levin *et al.* [1] formulates the colorization problem as a quadratic programming by assuming neighboring pixels with similar intensities should have similar colors. In [2], Yatziv *et al.* colorize an image by blending the given colors with weights related to the geodesic distances; while the work [3] minimizes the mixed l_0/l_1 norm to colorize images. In [4], Pang *et al.* seek for sparse representations that are consistent with both the luminance and the known chrominance.

Differently, Example-based colorization algorithms [5, 6, 7, 8] receive color cues in form of reference image(s) with similar colors. The works [5, 6, 7] take as input an example color image with similar textures and color "mood" to the target grayscale image. Welsh *et al.* [5] match the luminance and texture information between the images then transfer the colors onto the grayscale image; while the work [6] classifies the textures of the grayscale image in a feature space and transfers scattered color labels from the example image, then the image is colorized using [1]. And the work [7] resolves the colorization problem with a variational formulation. To colorize a grayscale image of a specific scene, Liu *et al.*'s work [8] accepts multiple example color images of the target scene with different lighting conditions, so as to achieves colorizations robust to illumination differences.

In this work, we propose a hybrid colorization scheme that combines the advantages of both categories. By incorporating the techniques used in example-based colorization methods into scribble-based methods, it is possible to better utilize the given chrominance information and further advance the colorization qualities of scribble-based methods. Our algorithm operates in YUV color space. Given a grayscale image (Y component) with a few color scribbles or scattered color labels indicating the desired chrominance on it, we aim at coloring the grayscale image faithfully via self-similarity on images and optimization techniques. Note that the property of self-similarity on natural images is adopted to address many image restoration problems (e.g., denoising [9], demosaicing [10], inpainting [11] and matting [12]). In the literature, algorithms exploiting self-similarity are also referred to as "non-local methods."

Our method firstly propagates the known chrominance based on self-similarity. It is achieved by searching for similar patches in

Part of this work was done when the first author was on an exchange in Tokyo Institute of Technology.

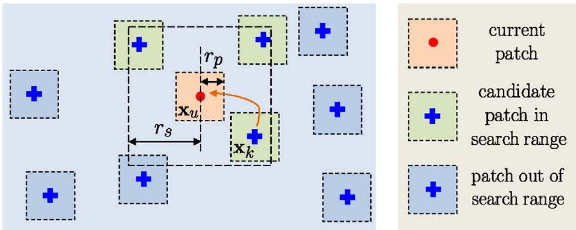


Fig. 2. Searching for a similar patch on the grayscale image. After the closest patch is found, color at the center of the similar patch is transferred to the center of current patch.

terms of luminance then transferring the given colors to locations where the colors are unknown. After that, we fully colorize the grayscale image with the given color scribbles (or scattered color labels) and the transferred chrominance obtained from the aforementioned step. By customizing Levin *et al.*'s work [1], we formulate the colorization problem as a quadratic programming problem. Our formulation regards only the pixels without given colors as the optimization variables and posts weighted soft constraints onto the colors transferred onto them. And the weights are determined by the similarities between the similar patches. In [13], Yao *et al.* also introduce a propagation step before colorization; however, they use a different propagation mechanism. And they treat both the given and propagated chrominance in the same way, which is inappropriate; and it also leads to a different formulation.

Fig. 1 presents the colorization procedure of our work. With the grayscale image and color scribbles on it (Fig. 1(a)), we propagate the given color information based on self-similarity (Fig. 1(b)), then the weights of the propagated chrominance are computed (Fig. 1(c)), where brighter intensities indicate larger weights. With the propagated colors and the input chrominance, our algorithm fully colorizes the grayscale image to obtain the result (Fig. 1(d)). Note that for all figures in this paper, we darken the given grayscale image and enhance the color cues for better display. We recommend to read the electronic version of this paper for the best views of the figures.

2. CHROMINACE PROPAGATION BASED ON SELF-SIMILARITY

Having received the grayscale image and the color scribbles (or scattered color labels), we firstly propagate the chrominance information based on the following two assumptions on self-similarity of images:

(i) for a natural image, there exist many similar patches on it in terms of intensities;

(ii) similar patches on an image tend to have similar colors, especially for edges and regular textures.

These two assumptions are "borrowed" from example-based colorization algorithms, where chrominance of the grayscale image is transferred from the example (or reference) color image(s).

Specifically, for an image patch $\mathbf{x}_u$ (subscript "u" means unknown) having size $(2r_p+1)$-by-$(2r_p+1)$ on the grayscale image centered on a pixel with unknown color, we search for another image patch $\mathbf{x}_k$ (subscript "k" means known) centered on a pixel with known color, where the search range is a square of size $(2r_s+1)$-by-$(2r_s+1)$ with the same center as $\mathbf{x}_u$; such that among all patches centered on a pixel with known color within the search range, $\mathbf{x}_k$ is the most similar to $\mathbf{x}_u$. The process of searching for a similar grayscale image patch is illustrated in Fig. 2.

In this work, we measure the similarity between two patches, say, $\mathbf{x}_u$ and $\mathbf{x}_k$, in a way as if computing the filter coefficients in bilateral filtering [14]. Denote the similarity as $s_{u,k}$, then

$$s_{u,k} = \exp\left(-\frac{\|\mathbf{H}(\mathbf{x}_u - \mathbf{x}_k)\|_2^2}{\sigma_r^2}\right) \cdot \exp\left(-\frac{\|\mathbf{d}_u - \mathbf{d}_k\|_2^2}{\sigma_s^2}\right), \quad (1)$$

where $\mathbf{x}_u, \mathbf{x}_k \in \mathbb{R}^{(2r_p+1)^2}$ are the vectorized grayscale patches, and $\mathbf{d}_u, \mathbf{d}_k \in \mathbb{R}^2$ are the 2-D coordinates of the patches $\mathbf{x}_u$ and $\mathbf{x}_k$, respectively. $\mathbf{H}$ is a $(2r_p+1)^2$-by-$(2r_p+1)^2$ diagonal matrix, whose diagonal entries come from a circular-symmetric Gaussian blurring kernel with standard deviation σ_d. With the aid of matrix $\mathbf{H}$, more emphases are placed onto the central area of the patches. Note that similarity measure (1) satisfies $0 < s_{u,k} < 1$ for two patches $\mathbf{x}_u$ and $\mathbf{x}_k$, and a higher $s_{u,k}$ indicates the two patches are more similar. After $\mathbf{x}_k$, the patch closest to $\mathbf{x}_u$, is found, we transfer the color (U and V components) at the center of $\mathbf{x}_k$ to the center of $\mathbf{x}_u$ (Fig. 2), so that the center pixel of $\mathbf{x}_u$ has its own color label. For a pixel whose color is unknown, in case there's not any known colors within its search range, we simply do not transfer any color onto it.

The aforementioned process of searching for similar patch and transferring the color is carried out for all grayscale patches centered on a pixel with unknown color. One may suspect that this search would bring a lot of computation; fortunately, since the candidate patches are limited to those lie within the search range and also centered on pixels with known colors, the cost of this search is quite acceptable. In this work, we set $r_p = 15$ and $r_s = 40$ for a reasonable tradeoff between complexity and colorization quality. As an example, Fig. 1(b) shows both the given colors and the propagated (namely, transferred) colors obtained from Fig. 1(a).

With the auxiliary of the propagated chrominance, we endeavor to colorize the grayscale image more reliably compared to those methods where only the given chrominance information is used in the colorization process (e.g., Levin *et al.*'s work [1]). However, unlike the given chrominance, the propagated color labels cannot be fully trusted. As a result, to colorize the image using optimization (will be introduced in Section 3), the U and V components of the pixels whose colors are unknown are regarded as the optimization variables. And we assign a weight (confidence) to each transferred color label indicating how trustworthy the transferred color is.

The weight of a transferred color label located at the center of a patch $\mathbf{x}_u$ is denoted as w_u. It is computed based on the similarity measure between $\mathbf{x}_u$ and its closest patch $\mathbf{x}_k$, namely, $s_{u,k}$. We define the weight w_u as follows,

$$w_u = \frac{s_{u,k}}{1+\exp(-\alpha(s_{u,k}-T))}, \quad (2)$$

which equals $s_{u,k}$ times a logistic function, so $0 < w_u < 1$ holds. The logistic function in (2) behaves like a smooth variant of hard thresholding, where T is a predefined threshold, while α controls the transition of the logistic. Here we choose $\alpha = 30$ and $T = 0.4$. Intuitively, if $s_{u,k}$ is too small, we cannot trust the transferred color from $\mathbf{x}_k$ at all and a w_u close to zero will be assigned. For a pixel whose color is unknown, in case there's no color transferred to it, its corresponding weight is set to be zero. The weights of all transferred colors together form a confidence map, like the one in Fig. 1(c).

3. COLORIZATION USING OPTIMIZATION WITH TRANSFERRED CHROMINANCE

Equipped with the propagated chrominance, this section elaborates the process of fully colorizing the grayscale image using convex op-

Fig. 3. Ten test images from the Kodak image collection. With the order of left to right then top to bottom, they are indexed as image 1, image 2, and so forth.

timization techniques. To facilitate the presentation, some notations are introduced. Suppose the input image has n pixels, among these n pixels, n_k of them have their desired colors given by the user, while n_u of them do not have known colors (some of these n_u pixels have transferred color labels obtained from the process of Section 2), and $n = n_k + n_u$. Our approach operates in YUV color space, as mentioned in Section 1. Moreover, the recovery of U and V components are carried out independently under the same rationale. So for simplicity, we only illustrate the recovery of the V component.

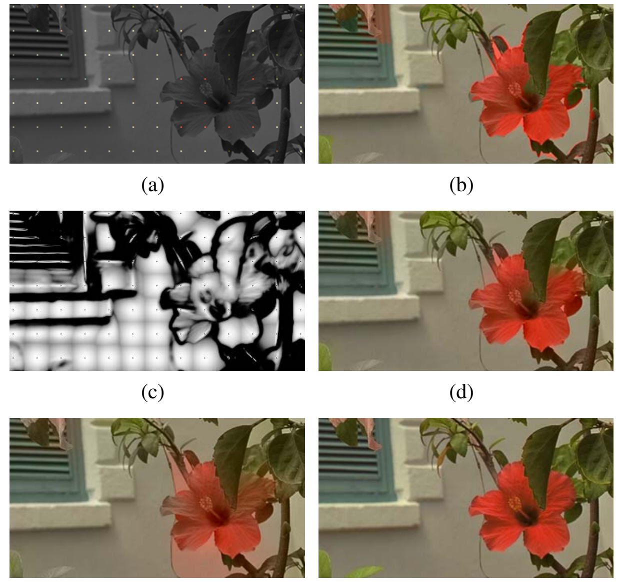
Suppose $\mathbf{v} \in \mathbb{R}^n$ is the vectorized V component of the image, its n elements can be separated into two vectors, $\mathbf{v}_k \in \mathbb{R}^{n_k}$ and $\mathbf{v}_u \in \mathbb{R}^{n_u}$. Here $\mathbf{v}_k$ contains all the known V components, while $\mathbf{v}_u$ contains all the unknown V components; our objective is to recover $\mathbf{v}_u$ reliably. We further define a vector $\mathbf{v}_t \in \mathbb{R}^{n_u}$ (subscript "t" means transferred) containing all the transferred V components, such that the i-th entries of $\mathbf{v}_t$ and $\mathbf{v}_u$ correspond to the same pixel whose color is unknown ($1 \leq i \leq n_u$). Note that an entry in $\mathbf{v}_t$ is set to be zero if its corresponding pixel doesn't have transferred color. After that, we construct a diagonal matrix $\mathbf{W} \in \mathbb{R}^{n_u \times n_u}$ whose diagonal elements are the weights (confidence) computed from (2). And the (i, i)-th entry of $\mathbf{W}$ is the weight corresponds to the transferred color of the i-th entry of $\mathbf{v}_t$ ($1 \leq i \leq n_u$). By concatenating $\mathbf{v}_u$ and $\mathbf{v}_k$, the vector $\left[\mathbf{v}_u^{\mathrm{T}}\ \mathbf{v}_k^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^n$ is obtained. To ease the formulation, we define a permutation matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$, such that $\mathbf{v} = \mathbf{P}\left[\mathbf{v}_u^{\mathrm{T}}\ \mathbf{v}_k^{\mathrm{T}}\right]^{\mathrm{T}}$. In other words, the entries of $\mathbf{P}\left[\mathbf{v}_u^{\mathrm{T}}\ \mathbf{v}_k^{\mathrm{T}}\right]^{\mathrm{T}}$ are arranged in the same order as that of the vectorized image.

For the continuation of our discussion, we take a closer look at the formulation of [1], where Levin *et al.* proposed the prior that nearby pixels with similar intensities should have similar colors. This prior can be expressed by the following energy function

$$E(\mathbf{c}) = \sum_{i=1}^{n} \Big(\mathbf{c}(i) - \sum_{j \in \mathcal{N}(i)} a_{i,j}\mathbf{c}(j)\Big)^2 , \qquad (3)$$

where i,j are indices of pixels and $\mathbf{c} \in \mathbb{R}^n$ is the vectorized U or V component to be recovered, and the coefficients $a_{i,j} \propto \exp\left(-(\mathbf{y}(i) - \mathbf{y}(j))^2/2\sigma_p^2\right)$. For a pixel i, $\sum_{j \in \mathcal{N}(i)} a_{i,j} = 1$ holds. Here $\mathbf{y}(i)$ denotes the luminance at pixel i and $\mathcal{N}(i)$ is the set of pixels in the neighborhood of i, σ_p is the variance of the luminance within the neighborhood. As can be seen, for two nearby pixels i and j with similar luminance, coefficient $a_{i,j}$ would be large. In our work, we construct a sparse matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ whose (i, j)-th entry is $a_{i,j}$. As a result, $\mathbf{A}$ behaves as a smoothing filter when right-multiplied by a vectorized image of size n. With aid of matrix $\mathbf{A}$, the energy function (3) can be rewritten as $E(\mathbf{c}) = \|\mathbf{c} - \mathbf{A}\mathbf{c}\|_2^2$. Note the neighborhood $\mathcal{N}(i)$ in our work is the 7×7 image patch centered on pixel i.

We further define a binary matrix $\mathbf{S} \in \mathbb{R}^{n_u \times n}$ that extracts (or selects) the n_u unknown colors from the vectorized image with an ordering the same as that of $\mathbf{v}_u$. Then by requiring:



(a) (b) (c) (d) (e) (f)

Fig. 4. Colorization of a region within image 3. (a) Luminance with sampled color labels. (b) Propagated colors. (c) Confidence map of the propagated colors. (d) Colorization of our method. (e) Colorization of method [1]. (f) Original color region of image 3.

(i) unknown colors are similar to the propagated colors to such an extent that is proportional to the confidence defined by (2);

(ii) nearby pixels with similar luminance values should have similar colors (same assumption as [1]);

we formulate the colorization problem as a quadratic programming,

$$\mathbf{v}_u^\star = \arg\min_{\mathbf{v}_u} \left\| \mathbf{SAP}\left[\mathbf{v}_u^{\mathrm{T}}\mathbf{v}_k^{\mathrm{T}}\right]^{\mathrm{T}} - \mathbf{v}_u \right\|_2^2 + \lambda \left\|\mathbf{W}(\mathbf{v}_u - \mathbf{v}_t)\right\|_2^2 . \qquad (4)$$

Here λ is the weighting parameter that controls the importance of the transferred colors. If $\lambda = 0$, (4) degrades to the formulation of [1], where the energy function (3) is minimized.

Note that problem (4) has closed-form solution. Since $\mathbf{SAP} \in \mathbb{R}^{n_u \times n}$, it can be written as $\mathbf{SAP} = [\mathbf{A}_u\ \mathbf{A}_k]$, where $\mathbf{A}_u \in \mathbb{R}^{n_u \times n_u}$ and $\mathbf{A}_k \in \mathbb{R}^{n_u \times n_k}$. Then (4) becomes

$$\mathbf{v}_u^\star = \arg\min_{\mathbf{v}_u} \|(\mathbf{I}_u - \mathbf{A}_u)\mathbf{v}_u - \mathbf{A}_k\mathbf{v}_k\|_2^2 + \lambda \left\|\mathbf{W}(\mathbf{v}_u - \mathbf{v}_t)\right\|_2^2 , \qquad (5)$$

where $\mathbf{I}_u \in \mathbb{R}^{n_u \times n_u}$ is an identity matrix. After that, the closed-form solution of $\mathbf{v}_u^\star$ can be obtained by taking derivative of the objective of (5) with respect to $\mathbf{v}_u$ then setting it to zero,

$$\begin{aligned} \mathbf{v}_u^\star = &\Big[(\mathbf{I}_u - \mathbf{A}_u)^{\mathrm{T}}(\mathbf{I}_u - \mathbf{A}_u) + \lambda\mathbf{W}^{\mathrm{T}}\mathbf{W}\Big]^{-1} \cdot \\ &\Big[(\mathbf{I}_u - \mathbf{A}_u)^{\mathrm{T}}\mathbf{A}_k\mathbf{v}_k + \lambda\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{v}_t\Big] . \end{aligned} \qquad (6)$$

Similarly, chrominance of the U component can be recovered with the propagated colors and the given chrominance information.

4. EXPERIMENTAL RESULTS

In this section, we demonstrate the colorization results of our work and compare them with those obtained by Levin *et al.*'s work [1].

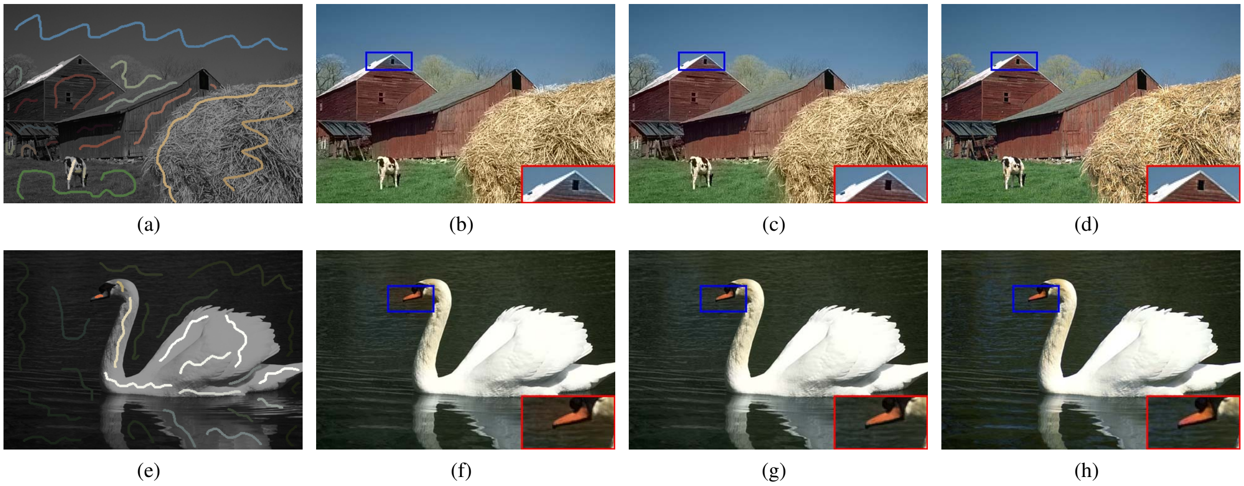


Fig. 5. Colorization with color scribbles as inputs. (a)(e) Grayscale images with color scribbles. (b)(f) Results of method [1]. (c)(g) Results of the proposed method. (d)(h) Original color images. Note the colors within the marked blue regions, their zoomed-in versions are also shown on the images within the red frames.

Table 1. CPSNR values of the 10 test images (in dB)

Image	1	2	3	4	5
Result of [1]	27.32	28.97	29.69	30.89	32.66
Proposed method	**29.13**	**30.27**	**30.91**	**32.00**	**33.48**
Image	6	7	8	9	10
Result of [1]	30.13	26.06	**26.93**	**32.05**	**30.54**
Proposed method	**30.81**	**26.43**	26.85	31.89	30.09

Note that for our work, the parameters are empirically chosen as: $\sigma_s = 30$, $\sigma_r = 3 \times 10^{-4}$, $\sigma_d = 3$ and $\lambda = 4 \times 10^{-3}$; and the luminance of the images are normalized to the range $[0, 1]$.

Ten test images from the Kodak PhotoCD [15] all having size of 512×768, as shown in Fig. 3, is used. We perform the following in this experiment. For each image, we uniformly sample one color pixel out of every 30×30 non-overlapping patches, which labels the color of of that pixel. As a result, about 99.9% of chrominance is removed. This sampling scheme, as shown in Fig. 4(a), may not be optimal for a particular image, we choose this typical sampling and apply it to all test images for its simplicity and representativeness. Then we colorize the images using the proposed method and method [1] using their provided implementation with the exact solver.

To quantitatively assess the colorization quality, we adopt the color peak-to-noise ratio (CPSNR) as the measurement. For two color images I_1 and I_2 of the same size $H \times W$, the CPSNR is

$$\mathrm{CPSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}},$$
$$\mathrm{MSE} = \frac{1}{3HW}\sum_{\Omega\in\{\mathrm{R,G,B}\}}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(I_1^{(\Omega)}(i,j) - I_1^{(\Omega)}(i,j)\right)^2, \tag{7}$$

where $I_1^{(\mathrm{R})}(i,j)$, $I_1^{(\mathrm{G})}(i,j)$ and $I_1^{(\mathrm{B})}(i,j)$ denote the R, G and B color components of the pixel locates at position (i,j) in the color image I_1, similar for I_2. We compute the CPSNRs between the colorizations and their corresponding ground-truth color images, then tabulate the results in Table 1. As can be observed, our method gives superior colorizations in terms of CPSNR compared to method [1].

Fig. 4 presents the colorizations of a region within image 3. As shown in Fig. 4(d), our result looks vivid and comparable to the original color image (Fig. 4(f)) even with very few color cues as inputs. Besides, note the flower and its surrounding region, our result gives more faithful colors compared to the result of [1] (Fig. 4(e)). This is because colors of a local region with edges and textures can be effectively transferred from another region similar to it. However, the process of color propagation may not always be beneficial if incorrect colors are transferred, that's why for images 8–10, our method generates results not so good as [1] (Table 1). This problem, which also exists in [6], is worthy of investigating for future work.

Our work accepts color scribbles or scattered color labels as input color cues. Fig. 5 demonstrates coloring two images from BSDS500 image database [16] with color scribbles as inputs. Note the marked blue regions, and their respective zoomed-in versions accompanied on the images. Compared to the results of Levin *et al.*'s method [1], which have severe color bleeding artifacts in the marked regions, our method gives more accurate and pleasant colorizations. As a result, our method better utilizes the known information, especially for colors on edges and regular textures.

5. CONCLUSION

The problem of image colorization is considered in this work. To address this highly ill-posed problem, we exploit the property of self-similarity on natural images and propose a hybrid colorization algorithm, which combines the advantages of scribble-based colorization algorithms and example-based colorization algorithms. Experimental results show that, the proposed method achieves state-of-the-art colorizations with very few input color cues.

6. ACKNOWLEDGMENT

This work is supported in part by Hong Kong Government Innovation and Technology Fund and the State Key Laboratory on Advanced Displays and Optoelectronics Technologies (Project No.: ITC-PSKL12EG02), and HKUST (HKUST Project No.: FS-GRF14EG40).

7. REFERENCES

[1] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH Conf.*, 2004, vol. 23, pp. 689–694.

[2] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. on Imag. Process.*, vol. 15, no. 5, pp. 1120–1129, 2006.

[3] K. Uruma, K. Konishi, T. Takahashi, and T. Furukawa, "Image colorization based on the mixed l0/l1 norm minimization," in *Proc. IEEE Int. Conf. Imag. Process.*, 2012, pp. 2113–2116.

[4] J. Pang, O. C. Au, K. Tang, and Y. Guo, "Image colorization using sparse representation," in *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Process.*, 2013, pp. 1578–1582.

[5] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 277–280, 2002.

[6] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Proc. 16th Eurograph. Symp. Rendering*, 2005, pp. 201–210.

[7] A. Bugeau, V.-T. Ta, and N. Papadakis, "Variational exemplar-based image colorization," *IEEE Trans. on Imag. Process.*, vol. 23, no. 1, pp. 298–307, 2014.

[8] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng, "Intrinsic colorization," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 152:1–152:9, 2008.

[9] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comp. Soc. CVPR*, 2005, pp. 60–65.

[10] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self-similarity driven color demosaicking," *IEEE Trans. on Imag. Process.*, vol. 18, no. 6, pp. 1192–1202, 2009.

[11] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. on Imag. Process.*, vol. 13, no. 9, pp. 1200–1212, 2004.

[12] Y. Shi, O.C. Au, J. Pang, K. Tang, W. Sun, H. Zhang, W. Zhu, and L. Jia, "Color clustering matting," in *IEEE Int. Conf. Multimedia and Expo*, 2013, pp. 1–6.

[13] C. Yao, X. Yang, J. Wang, L. Song, and G. Zhai, "Patch-driven colorization," *Optical Engineering*, vol. 49, no. 1, pp. 017001:1–017001:5, 2010.

[14] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th IEEE Int. Conf. Comp. Vis.*, 1998, pp. 839–846.

[15] R. Franzen, "Kodak lossless true color image suite," `http://r0k.us/graphics/kodak/`, 1999.

[16] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.