Palette-Based Image Recoloring Using Color Decomposition Optimization

Qing Zhang, Chunxia Xiao, *Member, IEEE*, Hanqiu Sun, and Feng Tang

***Abstract*— Previous works on palette-based color manipulation typically fail to produce visually pleasing results with vivid color and natural appearance. In this paper, we present an approach to edit colors of an image by adjusting a compact color palette. Different from existing methods that fail to preserve inherent color characteristics residing in the source image, we propose a color decomposition optimization for flexible recoloring while retaining these characteristics. For an input image, we first employ a variant of the *k*-means algorithm to create a palette consisting of a small set of most representative colors. Next, we propose a color decomposition optimization to decompose colors of the entire image into linear combinations of basis colors in the palette. The captured linear relationships then allow us to recolor the image by recombining the coding coefficients with a user-modified palette. Qualitative comparisons with existing methods show that our approach can more effectively recolor images. Further user study quantitatively demonstrates that our method is a good candidate for color manipulation tasks. In addition, we showcase some applications enabled by our method, including pattern colorings suggesting, color transfer, tissue staining analysis and color image segmentation.**

***Index Terms*— Image recoloring, palette, color decomposition.**

I. Introduction

MANIPULATING colors in photographs is an important problem with many applications in computer vision and image processing. In the past decades, there have been numerous works tackling this problem from different perspectives. Some approaches make complex color modifications possible by using an example image [1], [2], or propagating sparse user edits [3]–[5]. Others automatically adjust colors by leveraging color dataset [6] or using a semantic word depicting the desired style [7]. A recent work of Chang et al. [8] offers an appealing palette-based color manipulation tool, which unifies ease of use and range of expressiveness.

Though color transfer methods like that of Reinhard et al. [1] enable globally color modifications with an example image, an image exactly encoding the user-preferred changes is often unavailable in practice. Edit propagation methods allow users to selectively apply color modifications by annotating the input image with color strokes. Though this kind of method offers users a broad expressive range, unexpected results could be produced if the strokes are not drawn in their algorithm preferred manner. Thus, it actually requires user to understand how propagation essentially works, which introduces extra difficulty for user to learn and use. Style transfer methods [6], [7] make it much easier to adjust colors by using user-provided keywords. However, in many cases, simple keywords may fail to indicate the desired color modifications. Besides, the keywords yielding the desired result may be unpredictable.

Recent palette-based image recoloring methods [5], [8] start a new trend for color manipulation by demonstrating impressive results. They are easy to understand and use for novice users. Our approach belongs to this category. Despite the remarkable progress, these methods still have the limitation of producing unnatural results (see Fig. 8) when handling images with complex color distribution because they do not consider the essential relationship between the entire image and the color palette. Different from existing methods, in this paper we introduce a color decomposition optimization to decompose pixel colors into plausible combinations of the palette colors, which can represent all kinds of color characteristics in the source image. Our approach is able to produce results with vivid color and natural appearance.

Our approach is based on the assumption that the color of each pixel can be represented as a linear combination of a few basis colors in some feature space. Different from existing methods, we recolor an image (Fig. 1) via a color decomposition optimization that explores the underlying relationship between the pixels and the basis colors. As shown in Fig. 2, our approach to recoloring an image is comprised of three steps. To start with, we extract a few most representative colors that represent the full range of colors in the image to form a suitable palette. The palette not only provides a set of basis sources for subsequent color decomposition, but also serves as a concise interface for intuitive recoloring.

Manuscript received June 14, 2016; revised October 9, 2016 and December 28, 2016; accepted February 13, 2017. Date of publication February 19, 2017; date of current version March 9, 2017. This work was supported in part by the NSFC under Grant 61472288, Grant 61672390, Grant 61379087, and Grant 61602183, in part by NCET under Grant NCET-13-0441, in part by RGC research under Grant ref. 416212, in part by UGC under Grant 4055060, in part by the Fundamental Research Funds for the Central Universities 2042015kf0181 and in part by the State Key Laboratory of Software Engineering under Grant SKLSE-2015-A-05. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Q. Li. *(Corresponding author: Chunxia Xiao.)*

Q. Zhang is with the School of Computer, Wuhan University Wuhan 430072, China

C. Xiao is with the State Key Lab of Software Engineering and the School of Computer, Wuhan University, Wuhan, 430072 China (e-mail: zhangqing.whu.cs@gmail.com; cxxiao@whu.edu.cn).

H. Sun is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: hanqiu@cse.cuhk.edu.hk).

F. Tang is with Hewlett-Packard Laboratories, Palo Alto, CA 95014 USA (e-mail: sharp.tang@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2671779

Fig. 1. An example of our approach for image recoloring. Top-left: Original image (with source palette below). Others: different recoloring results corresponding to user-modified palettes (edited palette entries are underlined).

In the color decomposition step, we use a nonlinear energy minimization to compute a per-pixel linear combination of basis sources that explains the image. In the color manipulation step, users are allowed to edit the palette by replacing palette entries with their preferred colors. By recombining the user-modified palette with coding coefficients derived from the decomposition, we obtain the final recoloring results.

To decompose an image with M pixels, we have M equations and $N \times M$ unknowns (N denotes the number of basis colors in a palette). Thus, finding the linear combination that reasonably explains the colors of the whole image is an ill-posed problem, whenever the palette contains more than two basis colors. To address this underconstrained problem, the main task is to eliminate the ambiguities occurred during the decomposition. Therefore, we formulate the decomposition as an energy minimization problem. In particular, we incorporate a smoothness constraint into the decomposition to ensure that pixels with similar color be factored in a similar way. In addition, we apply a Total-Variation based sparseness prior to promote contributions from a sparse set of basis colors at each pixel. We demonstrate in a number of examples that our proposed color decomposition optimization enables us to produce high-quality image recoloring results with natural-looking appearance.

Compared with existing palette-based color manipulation methods, our approach has the following appealing characteristics: 1) it can always provide user instant feedback for any modifications to the palette while requires only once color decomposition as preprocessing. 2) it can recolor images without inducing visual artifacts encountered by existing methods. 3) it is applicable to various image editing applications.

In summary, our work has three main contributions:

- Introduced an automatic palette extraction approach, which can efficiently abstract the full range of colors in an input image to form a compact color palette.
- Developed color decomposition optimization for decomposing colors into plausible linear combinations with a given color palette, which enables effective image recoloring.
- Extended our method to several image editing applications, including pattern colorings suggesting, color transfer, tissue staining analysis and color image segmentation.

The rest of this paper is organized as follows: Section II reviews the related work. Section III describes how to summarize the main color groups of the input image for a palette. Section IV presents the technical details of the proposed color decomposition optimization. Section V gives the experimental results, comparisons and evaluations. Section VI introduces several image editing applications of our approach. Finally, we conclude the paper and discuss the future work in Section VII.

II. Related Work

Our work is related to color transfer, edit propagation, palette-based color manipulation and color decomposition. We review the related work from the four aspects in this section.

A. Color Transfer

Color transfer is a well-adopted technique for image recoloring, and has attracted much attention since the pioneering work of Reinhard et al. [1]. This method transfers colors from a reference image to the input image by applying a global color mapping to match statistics (mean and variation) between them. However, when the compatibility between the source and target images is low, it would produce unnatural results unless users manually construct reliable region correspondence by swatches. Other global approaches leverage histogram matching to transfer higher level statistical properties. Neumann and Neumann [9] achieved an exact match of the gamut of the target image by using 3D histogram matching in the HSL color space. Pitie et al. [2] estimated a continuous transformation that maps one N-dimensional distribution to another. This technique is effective in matching color palette between images since it takes the correlations between channels into consideration, but it tends to produce significant grain artifacts. Later, similar idea was reported in [10], [11] to remove this kind of artifact by preserving the gradient field of the source image with an optimization. By utilizing the midway histogram equalization, Papadakis et al. [12] proposed a variational model for histogram transfer of two or more color images.

To account for possible color mismatches between images, Tai et al. [13] presented a framework for regional color transfer using a modified expectation-maximization (EM) method which infers connectivity among pixels. Chang et al. [14] classified colors into perceptual categories via psychophysical experiments, and enabled automatic color transfer by matching colors within the same category. To make the algorithm more robust and able to handle video, an extension of this method is later introduced in [15]. Pouli and Reinhard [16] developed a histogram reshaping technique for transfer colors between images of arbitrary dynamic range by manipulating histograms at different scales. Recently, Hwang et al. [17] employed probabilistic Moving Least Squares (MLS) for color transfer.

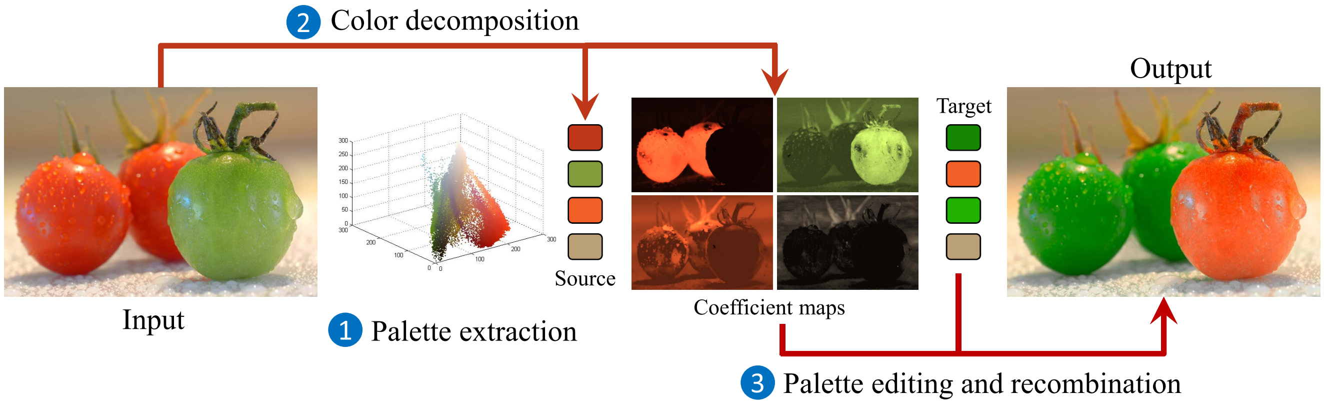


Fig. 2. The flowchart of our approach. Note that we visualize computed scalar coding coefficients in color decomposition step with corresponding basis colors in the source palette.

Nevertheless, all these methods require users to provide a reference image depicting the style they want as input.

B. Edit Propagation

Propagating sparse user edits to the whole image is another well-known technique for image recoloring. The first framework is proposed by Levin et al. [18] for colorizing grayscale images, which propagates user-supplied color strokes to nearby regions. Lischinski et al. [19] proposed to propagate tone edits in terms of strokes to entire image for local tonal adjustment. An and Pellacini [3] introduced "AppProp" where they computed all-pairs affinities between pixels for propagation. However, it would incur an intractable large linear system defined by a dense affinity matrix. Thus, they further employed the Nystr*ö*m method to approximate a low-rank matrix to reduce storage and computation.

Since edit propagation is usually formulated by quadratic energy minimization, it would be computationally expensive to solve linear systems induced by larger images or videos. Hence, various approaches are proposed to further accelerate edit propagation. Xu et al. [4] used a k-d tree in feature space to accelerate AppProp by working on clusters instead of individual pixels. Li et al. [20] reformulated edit propagation as a Radial Basis Function (RBF) interpolation problem rather than a global optimization for better performance. Xiao et al. [21] proposed a hierarchical approach using adaptive quadtree subdivision for acceleration. Bie et al. [22] presented an efficient edit propagation based on a sampling scheme. More recently, Chen et al. [5], [23] developed a sparsity-based edit propagation approach for high-resolution images and video by utilizing sparse dictionary learning.

Other approaches aim at improving the quality of edit propagation. Farbman et al. [24] replaced the Euclidean distance with diffusion distance for measuring affinities between pixels, which better accounts for the global distribution of pixels. Ma and Xu [25] avoided fringe-like artifacts commonly encountered by previous methods using anti-aliasing recovery. Later, Chen et al. [26] presented manifold preserving edit propagation, which reduces halo artifacts around boundaries by maintaining a manifold structure formed by all pixels in some feature space. This method was later sped up by [27], [28]. Xu et al. [29] proposed a sparse control model for edit propagation, which largely reduces the required amount of user interaction.

C. Palette-Based Color Manipulation

Recoloring an image by steering the color compositions in a palette leads a new trend in recent years. Wang et al. [6] proposed to modify colors of an image by matching a predefined color theme. However, they focused on addressing how to assign palette colors to different soft segments while our method seeks to explore the underlying relationship between each pixel and a palette abstracting the colors of the input image. A recent work of Lin et al. [30] described a method for automatically coloring 2D patterns by palettes based on a probabilistic factor graph model. More recently, Chang et al. [8] introduces a simple yet intuitive tool that allows users to recolor an image by editing a color palette. The approach is easy to understand and use while offering a broad expressive range. However, as demonstrated in our experiments, this method may produce results with unnatural or less vivid appearance for images with complex color distribution. Furthermore, this method would be not easy to understand since the final result would be unpredictable when multiple palette entries are simultaneously edited.

D. Color Decomposition

Decomposing colors into combinations of spectral signatures is an important problem in medical imaging. Rabinovich et al. [31] presented an unsupervised color decomposition method of histologically stained tissue samples for cancer diagnosis. Unlike our proposed color decomposition aiming at exploring the relationship between individual pixels and some representative basis colors, they seek to formulate it as a blind source separation problem on a stack of hyperspectral images by utilizing matrix factorization. To the best of our knowledge, our approach makes the first attempt to tackle a palette-based color decomposition problem.

III. Automatic Palette Extraction

As mentioned before, a suitable color palette not only facilitates reliable color decomposition, but also helps to present users an intuitive interface. In particular, a desirable palette is expected to have the following three properties. It should consist of a few most representative colors that abstract the main color categories in the source image. The correlation between any two palette colors should be as low as possible, namely palette colors should be far from each other. It should contain a moderate number of colors, not too few to be expressive enough and not too many to be easy of use.

Though there are methods [8], [32], [33] capable of creating a palette from an image in the literature, none of them can simultaneously meet all our requirements. Here we introduce a simple yet effective variant of k-means for extracting high quality palettes.

As naively clustering colors with k-means is computationally expensive even for medium sized image, for the sake of performance, we begin by reducing the number of colors in the input image. We first perform color quantization in RGB color space by assigning all colors to bins in a $16 \times 16 \times 16$ 3D histogram, which helps to reduce the number of colors to $16^3 = 4096$. Note that we compute the mean color in Lab color space for each bin because of its perceptual accuracy in measuring color differences. Each quantized color c_i now actually represents n_i pixels that assigned to that bin. We further reduce the number of colors by repeatedly removing the least frequently occurring color (namely color corresponds to the smallest n_i) under the constraint that remained colors should associate with pixels no less than 96% of the input image. In this way, we not only largely reduce the number of colors, but also ensure that subsequent operations are independent of the image size.

As pointed out by Pelleg and Moore [34], bad choice of initial centers for k-means typically incurs poor computational performance and local minimum. Thus, instead of randomly initializing the centers, we select k centers $\{\mu_i\}_{i=1}^{k}$ in a deterministic way [8]. Specifically, we choose the i-th center μ_i by

$$\mu_i = c_j \quad \text{s.t. } j = \arg\max_j \{n_j\} \tag{1}$$

In other words, each center is initialized as the color that currently represents the most pixels. To further ensure that the selected centers are far from each other, we attenuate the number of pixels n_i that each remained color c_i $(i \neq j)$ represents after each center selection by a factor

$$\ell_i = 1 - \exp(d_{ij}^2 / \delta_d^2), \tag{2}$$

where d_{ij} denotes the distance between color c_i and c_j in Lab color space, δ_d is set to 80 by default. Intuitively, colors that are closer to the previous selected center c_j would have larger falloffs in n_i, which reduces the likelihood of selecting a similar color in next center selection. We repeatedly perform above center selection until k centers have been selected. Then we use the k-means approach to obtain k color clusters, and compute weighted mean (weighted by original n_i for each color c_i) for each cluster.

As we analyzed, a moderate size of palette (determined by k) is also essential. Theoretically, a too small k may cause some colors might not be well-represented by combinations of the limited palette colors, while too large a k would promote

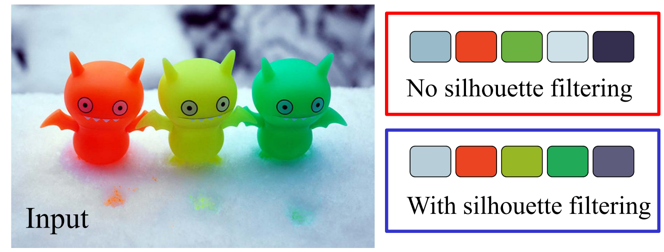


Fig. 3. Comparison on palette extraction with and without silhouette filtering.

Fig. 4. Comparison between our approach and other automatic palette extraction methods. (a)-(b) Result of O'Donovan et al. [32] and Lin et al. [33], which are slow (more than one minute) to generate the palettes. As the two methods assume a palette of size $k = 5$, we leave out their results in the third column as [8] do. (c) Result of Shapira et al. [35], which takes about 10 seconds to compute. (d) Result of a naively implementation of k-means algorithm, which is non-deterministic despite its advantage in efficiency (2 seconds). (e) Chang et al. [8] takes 60ms for extracting high-quality palettes. (f) Our method has similar computational performance as [8] while we produce palette with lower similarities between different color components. Testing images are from [8].

correlations between different palette colors and make it not easy to achieve a desired color editing result. Considering the color sparsity in natural images, we limit $k \in [3, 8]$ and find that $k = 5$ is a good default setting for color manipulation. However, users are allowed to choose k according to their needs. Fig. 3 shows an example, where we note that the method is biased towards merging two different color clusters at the price of neglecting certain clusters (yellow doll in this case). Thus, we introduce the following silhouette filtering to remove the bias.

To extract k palette colors free of the above mentioned bias, we begin by selecting 8 (the upper bound of the palette size) deterministic initial centers, and then perform k-means clustering with the 8 initial centers. If k is equivalent to 8, we directly take the k weighted means of clusters as palette colors. If k is smaller than 8, we first compute average silhouette value [36] for each cluster as a measure of how tightly and uniquely all the data in the cluster are. Then, we discard the initial center yielding the cluster with smallest silhouette value. Next, we cluster with the remained initial centers and further exclude the center corresponding to cluster with lowest silhouette value, repeating until the number of remained initial centers equals to k. Fig. 3 shows how the silhouette filtering improves the resulting palettes. In Fig. 4, we compare our approach with the previous palette extraction methods.

IV. Color Decomposition Optimization

According to the additive color mixing theory, adding red to green (blue) yields yellow (magenta). Similar to the idea behind the theory, our color decomposition optimization is based on the assumption that colors in the image can be represented as a per-pixel linear combination of some basis colors. Namely, we believe that the color of each pixel p can be approximately reconstructed as follows:

$$\boldsymbol{C}_p = \sum_{i=1}^{m} w_p^i \bar{\boldsymbol{C}}_i, \tag{3}$$

where $\boldsymbol{C}_p$ denotes a 2×1 color vector of pixel p consisting of its ab channels in Lab space, and w_p^i denotes a scalar coefficient measuring how much the color vector $\bar{\boldsymbol{C}}_i$ of the i-th basis color essentially contributes to the formation of $\boldsymbol{C}_p$. Note that the luminance channel L is not involved to avoid ambiguous decomposition caused by illumination difference.

To obtain the coding coefficients $\{w^i\}_{i=1}^m$ for an input image, we formulate the color decomposition problem as energy minimization of an objective function defined as follows:

$$\underset{\{w^i\}_{i=1}^m}{\arg\min}\, F = f_c + \lambda_n f_n + \lambda_s f_s + \lambda_r f_r, \tag{4}$$

where f_c, f_n, f_s and f_r are energy terms accounting for different constraints, while λ_n, λ_s and λ_r are all positive weights for balancing these terms in the objective function. We describe the details of these terms below.

A. Color Fidelity Term

According to our assumption, a valid decomposition should satisfy Equation (3), we thus define the following L_2 error metric to measure the reconstruction error:

$$f_c = \sum_p \left\| \boldsymbol{C}_p - \sum_{i=1}^{m} w_p^i \bar{\boldsymbol{C}}_p \right\|^2. \tag{5}$$

B. Non-Negativity Term

As negative coefficients are inexplicable and deviated from the additive color mixing theory, we enforce all the coefficients to be non-negative by penalizing the negative coefficients as follows:

$$f_n = \sum_i \sum_p \eta(w_p^i) \tag{6}$$

where $\eta(\cdot)$ refers to a function defined as

$$\eta(x) = \begin{cases} 0, & x \geq 0 \\ |x|, & x < 0 \end{cases} \tag{7}$$

Note that a simple hard bound constraint $w_p^i \geq 0$ can also work. For the sake of efficiency, we here use the soft constraint in Equation (6) instead, which produces similar results with less iterations.

C. Smoothness Term

Intuitively, similar colors should be decomposed in a similar way with approximately equivalent resulting coefficients. Considering the color similarities in a local window, we enforce smooth color decomposition with a smoothness term, which measures the L_1 norm of difference between coefficient w_p^i at pixel p and the weighted average of corresponding coefficients at the neighboring pixels:

$$f_s = \sum_i \sum_p \left| w_p^i - \sum_{q \in \Omega(p)} z_{pq} w_q^i \right|, \tag{8}$$

where $\Omega(p)$ refers to a 5×5 window centered at pixel p, z_{pq} is a weighting function measuring affinity between pixel p and q, which is defined as

$$z_{pq} = \frac{1}{Z_p} \exp(-\left\|\mathbf{f}_p - \mathbf{f}_q\right\|^2 / \delta_a) \exp(-\left\|\mathbf{x}_p - \mathbf{x}_q\right\|^2 / \delta_s) \tag{9}$$

where $\mathbf{f}_p$ is a Lab color vector at pixel p, and $\mathbf{x}_p$ is the spatial position (x and y coordinates) of pixel p. Z_p is the normalization constant. For all results in this paper, we set δ_a and δ_s to 500 and 10, respectively. In Equation (8), the L_1 norm is used to encourage sparsity, which can not only ensure smooth color decomposition among similar colors, but also helps to avoid smoothing out some color transitions.

D. Regularization Term

In fact, the combination of above energy terms tends to associate all basis colors during the color decomposition, which sometimes may not be the case. For instance, colors that are close to a certain basis color should receive significantly more contributions from this basis color rather than be affected by all basis colors. Thus, to encourage as few as possible basis colors contribute to the color at each pixel, and to avoid the singularity and numerical instability, we incorporate the following energy term into our objective function as a regularization term

$$f_r = \sum_i \left\| \phi^i w^i \right\|_1 = \sum_i \sum_p \varphi_p^i \left| w_p^i \right|, \tag{10}$$

where ϕ^i refers to a diagonal weighting matrix with φ_p^i on the diagonal, and we set φ_p^i to 1 in default.

E. Optimization

Minimizing Equation (4) actually involves an L_1 regularized convex optimization problem. The global minimum of this kind of problem can be obtained using iteratively reweighted least squares (IRLS) [37] or Split Bregman method [38]. Here, we choose to optimize F using the IRLS technique, which converts the problem into a series of tractable weighted least squares problems. Specifically, f_n, f_s and f_r can be

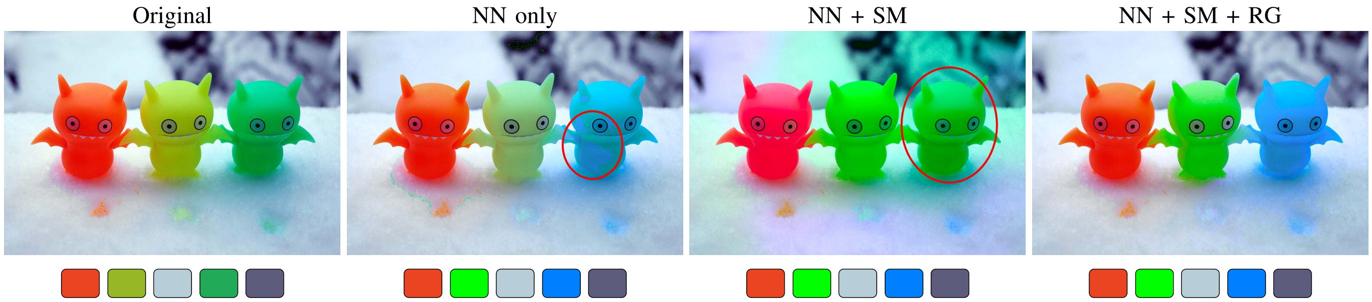


Fig. 5. Image recoloring using different energy terms of the color decomposition optimization. With data fidelity by default included, we gradually incorporate the non-negativity (NN), smoothness (SM) and regularization (RG) to validate the effectiveness of our color decomposition optimization. We note that the non-negativity term helps to obtain meaningful decomposition, but fails to ensure smooth color transition (**please zoom in on the red circle**). Due to the existence of ambiguities during decomposition, the result is not faithful to the user-modified palette. Further incorporating the smoothness term gives rise to smooth color transition among originally similar colors. However, the result yet fails to convey user's intent because of ignoring the sparsity in the contribution of each basis color. Finally, by incorporating the regularization term to encourage sparsity, we achieve natural looking result that responds faithfully to the user-modified palette.

reformulated as weighted quadratic terms as follows:

$$\hat{f}_n = \sum_i \sum_p \chi_p^i \left| w_p^i \right|^2 \tag{11}$$

$$\hat{f}_s = \sum_i \sum_p \xi_p^i \left| w_p^i - \sum_{q \in \Omega(p)} z_{pq} w_p^i \right|^2 \tag{12}$$

$$\hat{f}_r = \sum_i \sum_p \gamma_p^i \varphi_p^i \left| w_p^i \right|^2 \tag{13}$$

The weighting terms χ_p^i, ξ_p^i and γ_p^i are accordingly defined as

$$\chi_p^i = \begin{cases} 0, & w_p^i > 0 \\ (\left| w_p^i \right| + \varepsilon)^{-1}, & otherwise \end{cases} \tag{14}$$

$$\xi_p^i = (\left| w_p^i - \sum_{q \in N(p)} z_{pq} w_p^i \right| + \varepsilon)^{-1} \tag{15}$$

$$\gamma_p^i = (\left| w_p^i \right| + \varepsilon)^{-1} \tag{16}$$

where ε is a small constant used for preventing division by zero. By substituting above weighted square terms into Equation (4), we can obtain the following quadratic objective function, which has a unique global minimum and can be efficiently solved in a closed form.

$$\hat{F} = f_c + \lambda_n \hat{f}_n + \lambda_s \hat{f}_s + \lambda_r \hat{f}_r \tag{17}$$

To obtain the global minimum of F, the IRLS optimization would alternate between updating the weights and solving the weighted least square problem $\hat{F}$ until convergence. The detailed procedure for the color decomposition optimization is summarized in Algorithm 1.

F. Parameter Settings

Intuitively, non-negativity is a strong constraint. While smoothness and regularization should have roughly equal contributions to a plausible decomposition, and are obviously less important than non-negativity. Based on this observation, we have experimentally found that $\lambda_n = 10^3$, $\lambda_s = 1$ and $\lambda_r = 1$ works well in practice. For the number of iterations τ, we fix it to 8, which is proved to be sufficient for convergence. In Fig. 5, we evaluate effect on color manipulation of different energy terms in the color decomposition optimization.

Algorithm 1 Color Decomposition Optimization

Input: Image I, balancing weights λ_n, λ_s, λ_r, number of iterations τ
Output: Resulting coefficients $\{w^i\}_{i=1}^m$
Compute color palette $\{\bar{C}_i\}_{i=1}^m$ for image I.
Initialize χ_p^i, ξ_p^i, γ_p^i to be all 1, and t to 0.
repeat
  With χ_p^i, ξ_p^i and γ_p^i, solve Equation (17) for $\{w^i\}_{i=1}^m$.
  With $\{w^i\}_{i=1}^m$, update χ_p^i, ξ_p^i and γ_p^i.
  $t \leftarrow t + 1$
until $t \geq \tau$

G. Acceleration

Since the proposed color decomposition optimization requires iteratively solving a weighted linear system, it would be computationally expensive to perform color decomposition on high-resolution images. Here, we introduce an acceleration scheme that enables more efficient and scalable color decomposition on high-resolution images. Specifically, we first downsample the original image by factor of $\lfloor max(width, height)/200.0 \rfloor$ and then perform color decomposition on the downsampled image. Finally, we use joint bilateral upsampling (JBU) [39] to upsample the obtained coefficients to the full original resolution of the input image. In this way, we can avoid inducing degraded boundaries to final results because both the decomposition and color manipulation involve two chrominance channels *ab* only, without altering the luminance channel L all along.

With this accelerated implementation, we can not only obtain visually similar recoloring result (Fig. 6) to that of a naive implementation, but also reduce the time consumption for decomposing this 1024×1024 color image from 8 minutes to 2 seconds. Note that our experiments run on a laptop with a 2.4GHz Intel Core 2 Duo CPU.

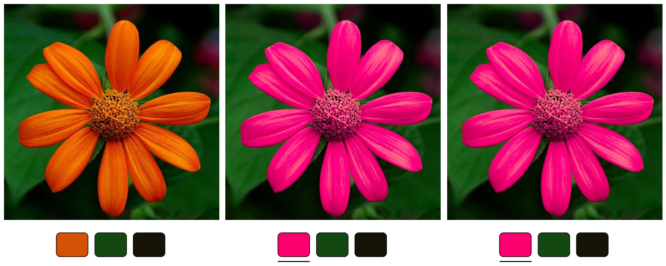


Fig. 6. Acceleration. Left to right: original image, results derived from naive implementation and the accelerated version, respectively.

Fig. 7. Video recoloring on the Girl[1] sequence. Left and right are input and output, respectively. We here make the girl's shirt green using the extension of our approach for video sequence. **Please refer to the accompanying video for more complete video sequence**.

H. Extension to Video

To recolor a video, we first sample some equally spaced video frames. Then, we compute an average palette that approximately abstracts the representative colors of the entire video from these sampled frames. Based on the palette, we perform the color decomposition optimization on each sampled frame. Thereafter, we propagate the decomposition on each sampled frame to subsequent frames within the sampling interval, by utilizing a probabilistic inference approach [40]. Fig. 7 shows an example of video recoloring.

V. Experimental Results

In this section, we evaluate our method from three aspects. We first compare our approach with other palette-based image recoloring methods. Then, we conduct a user study to further validate the usability of our approach. Finally, we examine the performance of our method in processing low light level and noisy scenes.

Currently, evaluating palette-based recoloring algorithm remains a challenge because it involves judgement of personal preference, and there is no consensus benchmark or standard dataset in the literature. To better evaluate our approach, by automatically grabbing images from Flickr on some keywords (color, landscape, flower, toy, car, fruit, sport, etc.), we collect 500 color images varying in scenes, styles and lighting conditions as candidate testing images.[2]

A. Qualitative Comparisons

In this experiment we demonstrate that our method can avoid inducing unpleasant visual artifacts encountered by the existing methods. To this end, we compare our approach with the state-of-the art palette-based recoloring methods [5], [8] on images randomly selected from the candidate testing images. Note that we do not make comparisons to color transfer methods [1], [2], [16] because they all require an additional example image as input. To avoid unfair comparison, we prepare same source palette and user-modified palette for all methods participated in this experiment. Specifically, we use our automatic palette extraction algorithm to compute high quality palettes for all sampled images, which helps to further exclude the possible impact of poor source palettes on final recoloring results.

Fig. 8 shows comparisons to [5] and [8]. As can be seen, our approach generates more vivid recoloring results that react faithfully to the user-modified palettes. Though [5] provides a fast and space efficient solution to the palette-based recoloring task, it may fail to handle regions with complex color transitions, such as the untouched hair ends in the second row and the unreal attached shadows in the third row. Results of [8] obtained with their public demo program[3] indicate that their approach tends to produce results with unnatural appearance, despite its efficiency. For instance, we can notice obvious local contrast reduction in their results shown in the third row (see attached shadows and drapes on the kimono), and in the fifth row (see the bottom right corner of the sweater). Besides, their results look less vivid while our results present more natural-looking appearance.

B. User Study

Here we conduct a user study to further validate the fact that our approach is a better choice for color manipulation tasks.

We start by inviting 5 volunteers to randomly select 50 images from the candidate testing images (10 for each volunteer). Then they are asked to write down expected color manipulations for their selected images in a text description manner like "change the yellow car to green" or "turn the orange fish to blue". Later, we find two proficient Photoshop users to create 50 target images for the 50 randomly selected original images based on the text descriptions. In order to avoid possible personal biases, we invite another 15 volunteers to adjust colors of the assigned originals to match the targets, using our method and two others.

[1] http://media.au.tsinghua.edu.cn/yegenzhi/IntrinsicVideo.htm

[2] We release all images at: https://github.com/CodeQingZhang/CIR

[3] http://recolor.cs.princeton.edu/demo/index.html

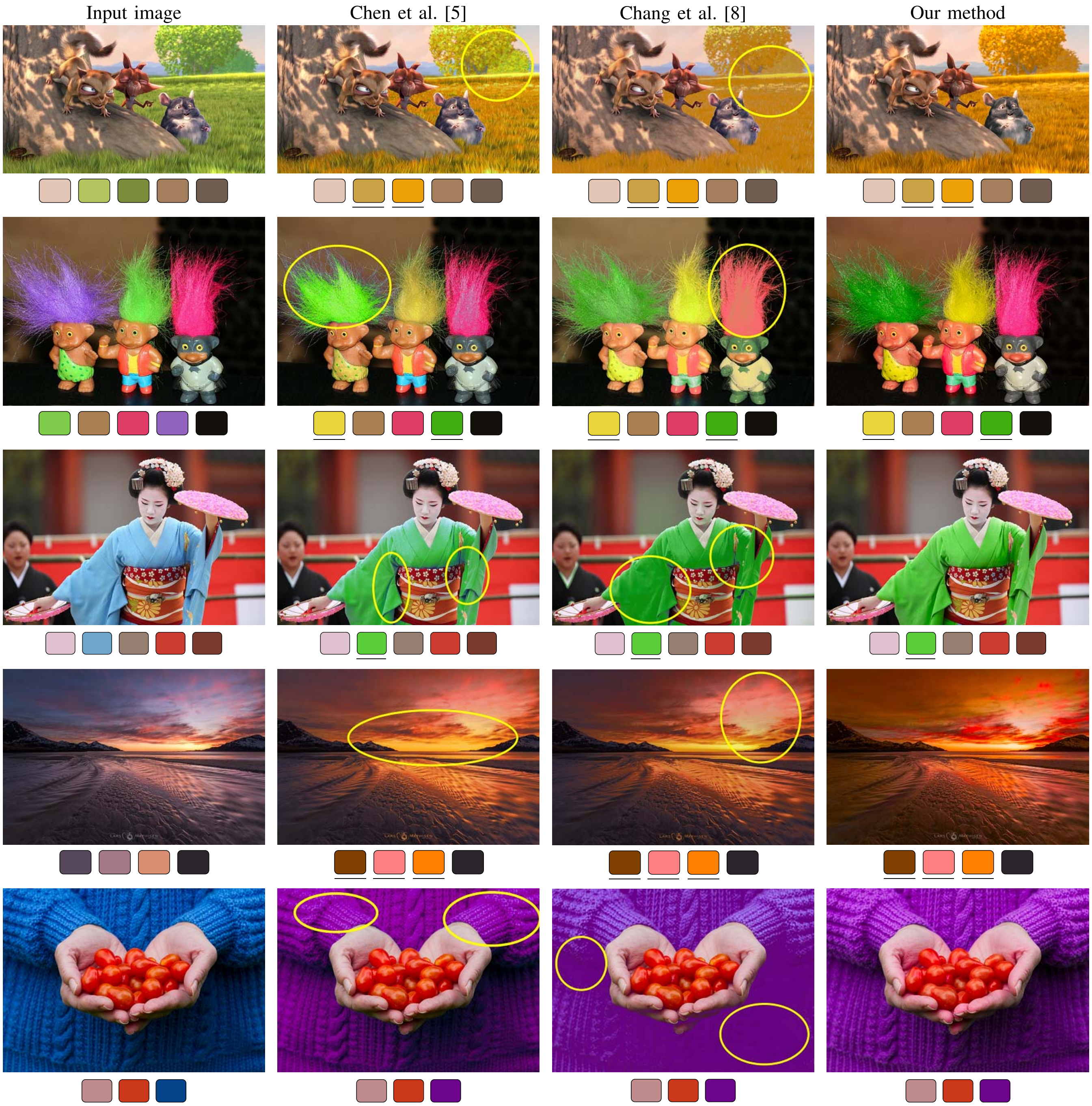


Fig. 8. Image recoloring results for examples randomly selected from the candidate testing images. Left to right: original image, results of [5], [8] and our method. **Please zoom in on the yellow circles to catch the details.**

We compare our method with "Replace Color" feature in Photoshop CS6 and the state-of-the-art palette-based recoloring method introduced in [8]. As a precondition for the experiment, we give all volunteers a brief introduction about how to use our method and the other two. As there are three candidates (our method and two others) for each original, we actually have 150 matching tasks in total. To avoid the subjective bias, we divide the 15 volunteers evenly into 5 groups, and assign each group with 10 original-target pairs. For each original-target pair, we randomly assign the three corresponding matching tasks to three different volunteers in that group. In this way, we ensure that there is no secondary edit to the same original for each volunteer. Besides, to prompt each volunteer to create the best result for each method, we inform them to try their best to edit the original to match the target in the shortest possible time. Once they are satisfied with their recoloring results, they can step into the next assigned matching task. Meanwhile, we record the time that each method takes to complete a task.

We collect 150 recoloring results (50 for each method), and also task completion time from the user study. Fig. 9 shows some examples. Intuitively, a method that produces the most

Fig. 9. Several examples from the user study. Left to right: original, target, results of the "Replace Color" feature in Photoshop CS6, method [8] and our approach.

similar result to the target in the least time has the highest usability, which means that we can evaluate the usability of the methods participated in the user study in terms of result quality and time consumption.

In order to quantitatively assess the result quality, namely how well each result matches with the corresponding target, we calculate the root-mean-square error (RMSE) of all the user study results for comparison. We choose the CIEDE2000 [41] metric, which is superior to others in measuring small color difference. The RMSE of each result can be calculated by

$$RMSE(I) = \sqrt{\frac{1}{N}\sum_{p}\left\|D(I_p, T_p)\right\|^2}, \tag{18}$$

where I is the color editing result, T is the corresponding target image, N is the number of pixels within the image I, p refers to a pixel, and $D(.)$ is a function that computes the CIEDE2000 color difference between two colors. Apparently, low RMSE indicates that the result is satisfying and perceptually close to the target, while a high RMSE means that the result is perceptually dissimilar to the target.

In Fig. 10, we show the RMSEs for all results in the user study. As can be observed, results of the "Replace Color" feature correspond to the overall highest RMSEs because the method is sensitive to illumination difference, and fail to propagate color edits to regions with inhomogeneous illumination. Chang et al. [8] exhibits better performance than the "Replace Color" feature. Since our method is independent of illumination difference and able to avoid visual artifacts encountered by [8], it achieves the lowest RMSEs in most cases. Though our method fails to exactly match the target generated by proficient Photoshop user, it allows novice users to produce comparable results, as shown in Fig. 9.

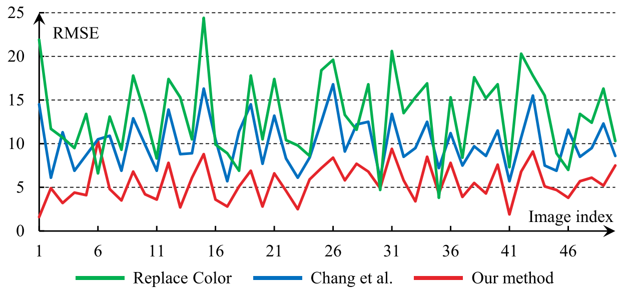


Fig. 10. Comparing RMSE of CIEDE2000 distance across the three methods in our user study.

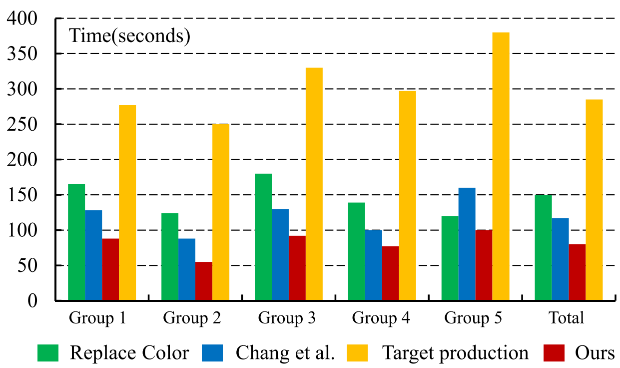


Fig. 11. Average time consumption for each method to fulfil a matching task in the user study.

Fig. 11 shows the average time that each method takes to complete a matching task by group. We also show time for proficient Photoshop user to produce a target as reference. As [8] may induce unnatural appearance, it takes users extra time to achieve the satisfied results. The "Replace Color" feature is less efficient than [8] because it requires users to manually specify the entire region to impose the edits, which is actually nontrivial for novice users. Overall, our approach takes the least time to complete an assigned recoloring task in most cases. Compared with the three methods in the user study, it takes more time for proficient Photoshop user to produce a target.

Above experiments show that our method produces the results with overall highest quality (lowest RMSEs) and minimum time consumption in the user study, which demonstrates that our method has better usability than the other two. Though our method is not as powerful as professional software such as Photoshop, it is well suited for recoloring images with relatively lightweight color manipulation tasks. Besides, it is easy to understand and use, which we think is also important for a color manipulation tool. Thus, we believe that our approach can be a good candidate for novice users to recolor their photos.

C. Low Light Level and Noisy Scenes

As shown in the top row of Fig. 12, our method is robust enough to low light scenes since the proposed color decomposition approach is independent of the scene illumination. However, we actually do not recommend users to process poor illuminated images with our method because these images often lead to a series of very dark palette colors. As dark colors are typically hard to distinguish, it would be nontrivial for novice users to perform color editing. To facilitate edits to images of low light level, users can firstly improve the visibility by performing illumination recovery [42], [43].

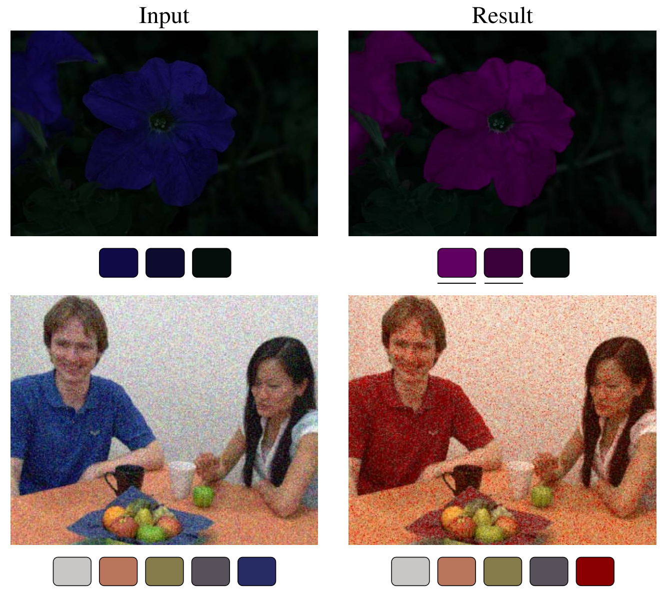


Fig. 12. Image recoloring results on low light level and noisy scenes.

For input image with a substantial amount of noise in the bottom row of Fig. 12, our method can also effectively recolor the man's shirt. A byproduct is that colors of noise are edited as well. Apart from this, our method has neither induced other artifacts nor amplified the magnitude of the noise. For images with heavy noise, we suggest user employ image denoising techniques [44], [45] to remove the noise first. Our method can then be applied to recoloring the image.

VI. Applications

Here we show several image editing applications to demonstrate the scalability of our approach.

A. Pattern Colorings Suggesting

Colored patterns play important roles in many fields, ranging from graphic/fashion design and manufacture to interior decoration. However, even for experienced artists, creating visually pleasing pattern colorings usually takes much time and manual efforts. Here, we adapt our approach to an automatic pattern coloring suggesting process, which can generate various colorings for a given colored pattern and enable users to select their preferred pattern colorings.

For an input colored pattern, we first compute a color palette and perform color decomposition with the palette. We then construct a palette database by screening popular palettes from online repositories (such as COLOURLovers and Adobe Kuler) of color palettes. By replacing the original palette with permutation of different palettes from the database, we can suggest users a diverse set of pattern colorings for personalized selection, as shown in Fig. 13.

Fig. 13. Pattern colorings suggesting. Original patterns are on left, and are followed by four different pattern colorings.

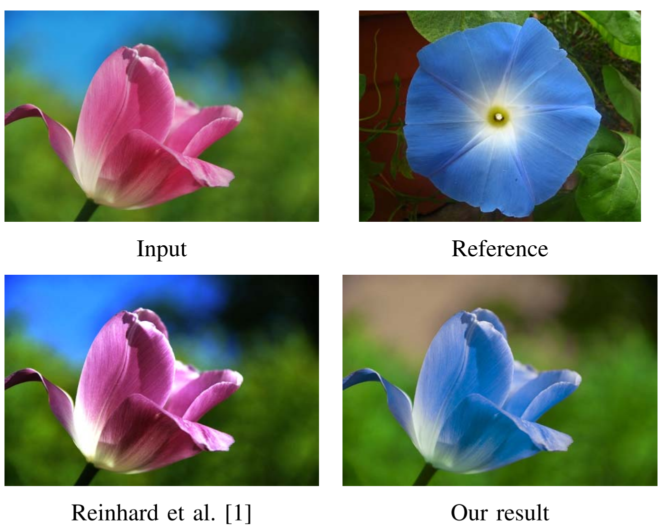


Fig. 14. Color transfer. Our method successfully transfer colors between the two flowers, while Reinhard et al. [1] just enhance the tone and contrast.

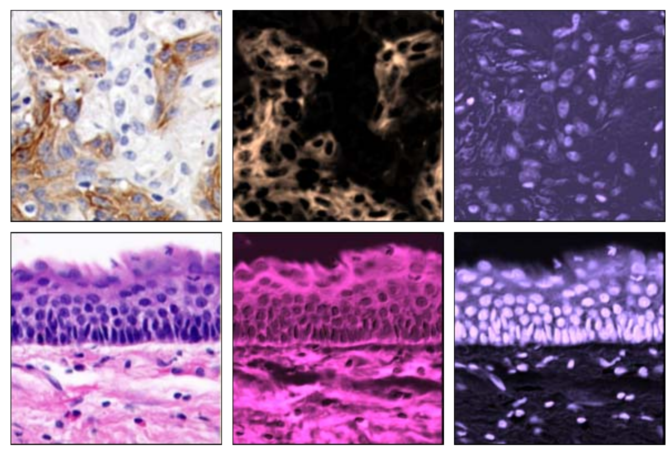


Fig. 15. Tissue staining analysis. **The first column**: input histologically stained images, the top one is stained with Haematoxylin and DAB, the bottom one with Haematoxylin and Eosin. **The second and third column**: separated component images for different dyes.

B. Color Transfer

Given a reference image, our method can transfer color and tone of the reference image to a target image by assigning the palette colors of the reference image to the target image. Compared with traditional statistics based color transfer method [1] (Fig. 14), we can achieve more flexible and effective color transfer because our approach enables users to selectively transfer the colors by controlling the color assignment.

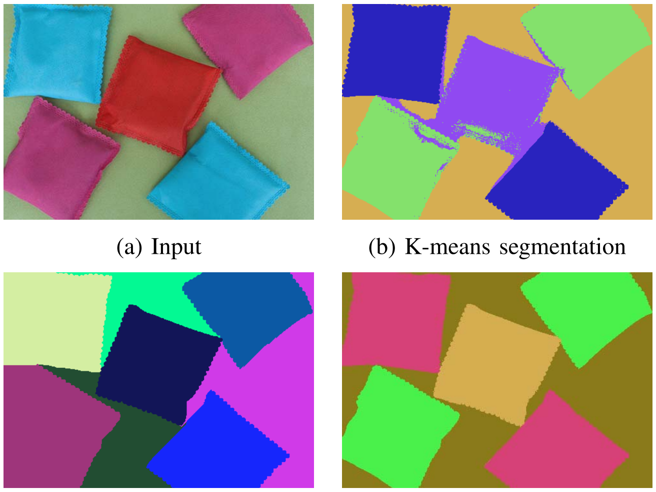
C. Tissue Staining Analysis

Currently, cancer is mainly diagnosed by means of visual examination under a microscope of tissue sections from biopsies. As pathologists rely on different tissue stains to identify morphological features, tissue staining analysis is becoming increasingly important for reliable cancer diagnosis. To achieve automatic tissue recognition via color separation, tissue staining analysis in terms of non-blind color decomposition of histological images is essential.

Our approach can provide reliable tissue classification as follows. For an input histologically stained image, we first construct a color palette according to the dyes used in staining process. For example, it is known in advance that hematoxylin stains cell nuclei blue. Then, we perform color decomposition with the palette. Fig. 15 shows two examples. As can be observed, our method provides good color separations on stained tissues, which would benefit accurate identification of morphological feature that may be linked to cancer.

D. Color Image Segmentation

Our approach can also be used for color image segmentation by utilizing coefficient maps derived from the color decomposition optimization. Since our color decomposition is independent of the luminance channel, it is more robust to changes in illumination, as demonstrated in Fig. 16.



(a) Input (b) K-means segmentation

(c) Graph-based segmentation (d) Our result

Fig. 16. Color image segmentation. (a) Input. (b) Result of k-means implementation included in OpenCV 2.4.11. (c) Result of graph-based segmentation [46]. (d) Our result.

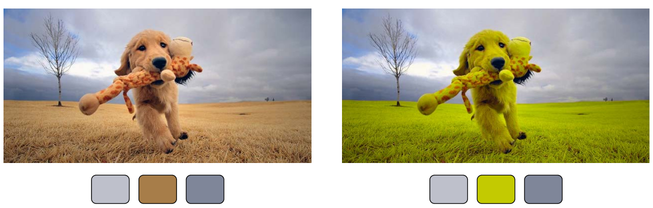


Fig. 17. Failed case. Left to right: input and our color editing result.

VII. Conclusion

We have presented an approach for recoloring images based on a color palette. We first apply a variant of k-means algorithm to summarize colors in the input image into a compact palette. By regarding colors in the palette as basis sources, we perform the proposed color decomposition optimization to obtain a per-pixel linear combination of the basis sources that reasonably explains the image. Then, we enable users to recolor the image by recombing the computed coding coefficients with a user-modified palette. Experimental results have shown that our approach outperforms existing methods. Further user study indicates that our approach can be a good choice for color manipulation tasks.

Our approach has limitations. Since it dose not take scene semantics into consideration, modifying the palette entries sometimes may not exactly convey the user's intent. As shown in Fig. 17, we plan to recolor the grassland, but induce unwanted changes to the puppy as a consequence. In future work, we would like to incorporate the scene semantics into our method to constrain which region to edit. Another limitation is that our current method still can not work in real time. We are interested in investing alternate optimization strategies that would allow more efficient color edits to high-resolution images or video. Extending our approach to large images with many different objects is also a promising direction to further improve the practicality.

Acknowledgment

The authors would like to thank all the anonymous reviewers for their insightful comments and constructive suggestions. This work was partly supported by the NSFC (No. 61472288, 61672390, 61379087, 61602183), NCET (NCET-13-0441), RGC research grant (ref. 416212), UGC grant (no. 4055060), the Fundamental Research Funds for the Central Universities (2042015kf0181) and the State Key Lab of Software Engineering (SKLSE-2015-A-05). Chunxia Xiao is the corresponding author.

References

[1] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Sep./Oct. 2001.

[2] F. Pitie, A. C. Kokaram, and R. Dahyot, "N-dimensional probability density function transfer and its application to color transfer," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Oct. 2005, pp. 1434–1439.

[3] X. An and F. Pellacini, "AppProp: All-pairs appearance-space edit propagation," *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008, Art. no. 40.

[4] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu, "Efficient affinity-based edit propagation using K-D tree," *ACM Trans. Graph.*, vol. 28, no. 5, Dec. 2009, Art. no. 118.

[5] X. Chen, D. Zou, J. Li, X. Cao, Q. Zhao, and H. Zhang, "Sparse dictionary learning for edit propagation of high-resolution images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2854–2861.

[6] B. Wang, Y. Yu, T.-T. Wong, C. Chen, and Y.-Q. Xu, "Data-driven image color theme enhancement," *ACM Trans. Graph.*, vol. 29, no. 6, Dec. 2010, Art. no. 146.

[7] Y. Liu, M. Cohen, M. Uyttendaele, and S. Rusinkiewicz, "AutoStyle: Automatic style transfer from image collections to users' images," *Comput. Graph. Forum*, vol. 33, no. 4, pp. 21–31, 2014.

[8] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring," *ACM Trans. Graph.*, vol. 34, no. 4, p. 139, 2015.

[9] L. Neumann and A. Neumann, "Color style transfer techniques using hue, lightness and saturation histogram matching," in *Proc. 1st Eurograph. Conf. Comput. Aesthetics Graph., Vis. Imag.*, May 2005, pp. 111–122.

[10] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Comput. Vis. Image Understand.*, vol. 107, nos. 1–2, pp. 123–137, Jul./Aug. 2007.

[11] X. Xiao and L. Ma, "Gradient-preserving color transfer," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1879–1886, Oct. 2009.

[12] N. Papadakis, E. Provenzi, and V. Caselles, "A variational model for histogram transfer of color images," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1682–1695, Jun. 2011.

[13] Y.-W. Tai, J. Jia, and C.-K. Tang, "Local color transfer via probabilistic segmentation by expectation-maximization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Jun. 2005, pp. 747–754.

[14] Y. Chang, S. Saito, K. Uchikawa, and M. Nakajima, "Example-based color stylization of images," *ACM Trans. Appl. Perception*, vol. 2, no. 3, pp. 322–345, Jul. 2006.

[15] Y. Chang, S. Saito, and M. Nakajima, "Example-based color transformation of image and video using basic color categories," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 329–336, Feb. 2007.

[16] T. Pouli and E. Reinhard, "Progressive color transfer for images of arbitrary dynamic range," *Comput. Graph.*, vol. 35, no. 1, pp. 67–80, Feb. 2011.

[17] Y. Hwang, J.-Y. Lee, I. S. Kweon, and S. J. Kim, "Color transfer using probabilistic moving least squares," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3342–3349.

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

[19] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 646–653, Jul. 2006.

[20] Y. Li, T. Ju, and S.-M. Hu, "Instant propagation of sparse edits on images and videos," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2049–2054, 2010.

[21] C. Xiao, N. Yongwei, and T. Feng, "Efficient edit propagation using hierarchical data structure," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 8, pp. 1135–1147, Aug. 2011.

[22] X. Bie, H. Huang, and W. Wang, "Real time edit propagation by efficient sampling," *Comput. Graph. Forum*, vol. 30, no. 7, pp. 2041–2048, Sep. 2011.

[23] X. Chen, J. Li, D. Zou, and Q. Zhao, "Learn sparse dictionaries for edit propagation," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1688–1698, Apr. 2016.

[24] Z. Farbman, R. Fattal, and D. Lischinski, "Diffusion maps for edge-aware image editing," *ACM Trans. Graph.*, vol. 29, no. 6, Dec. 2010, Art. no. 145.

[25] L.-Q. Ma and K. Xu, "Efficient antialiased edit propagation for images and videos," *Comput. Graph.*, vol. 36, no. 8, pp. 1005–1012, Dec. 2012.

[26] X. Chen, D. Zou, Q. Zhao, and P. Tan, "Manifold preserving edit propagation," *ACM Trans. Graph.*, vol. 31, no. 6, Nov. 2012, Art. no. 132.

[27] L.-Q. Ma and K. Xu, "Efficient manifold preserving edit propagation with adaptive neighborhood size," *Comput. Graph.*, vol. 38, pp. 167–173, Feb. 2014.

[28] P. Musialski, M. Cui, J. Ye, A. Razdan, and P. Wonka, "A framework for interactive image color editing," *Vis. Comput.*, vol. 29, no. 11, pp. 1173–1186, Nov. 2013.

[29] L. Xu, Q. Yan, and J. Jia, "A sparse control model for image and video editing," *ACM Trans. Graph.*, vol. 32, no. 6, Nov. 2013, Art. no. 197.

[30] S. Lin, D. Ritchie, M. Fisher, and P. Hanrahan, "Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs," *ACM Trans. Graph.*, vol. 32, no. 4, Jul. 2013, Art. no. 37.

[31] A. Rabinovich, S. Agarwal, C. A. Laris, J. H. Price, and S. Belongie, "Unsupervised color decomposition of histologically stained tissue samples," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 667–674.

[32] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Color compatibility from large datasets," *ACM Trans. Graph.*, vol. 30, no. 4, Jul. 2011, Art. no. 63.

[33] S. Lin and P. Hanrahan, "Modeling how people extract color themes from images," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, Apr./May 2013, pp. 3101–3110.

[34] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proc. ICML*, vol. 1. Jun./Jul. 2000, pp. 727–734.

[35] L. Shapira, A. Shamir, and D. Cohen-Or, "Image appearance exploration by model-based navigation," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 629–638, Apr. 2009.

[36] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

[37] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Commun. Statist.-Theory Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[38] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[39] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, p. 96, Jul. 2007.

[40] G. Ye, E. Garces, Y. Liu, Q. Dai, and D. Gutierrez, "Intrinsic video and applications," *ACM Trans. Graph.*, vol. 33, no. 4, Jul. 2014, Art. no. 80.

[41] M. R. Luo, G. Cui, and B. Rigg, "The development of the CIE 2000 colour-difference formula: CIEDE2000," *Color Res. Appl.*, vol. 26, no. 5, pp. 340–350, Oct. 2001.

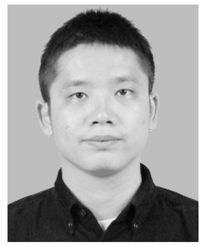
[42] L. Zhang, Q. Zhang, and C. Xiao, "Shadow remover: Image shadow removal based on illumination recovering optimization," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4623–4636, Nov. 2015.

[43] Q. Zhang, Y. Nie, L. Zhang, and C. Xiao, "Underexposed video enhancement via perception-driven progressive fusion," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 6, pp. 1773–1785, Jun. 2016.

[44] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

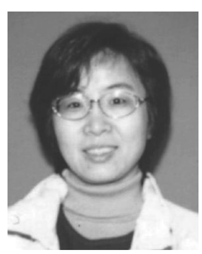
[45] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, Apr. 2012.

[46] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

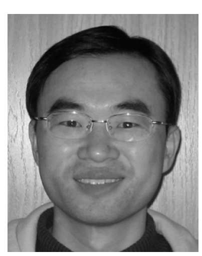


Qing Zhang received the B.Sc. and M.Sc. degrees in computer science and technology from Wuhan University in 2011 and 2013, respectively, where He is currently pursuing the Ph.D. degree with the School of Computer. His research interests include image and video processing.

Chunxia Xiao (M'10) received the B.Sc. and M.Sc. degrees from the Mathematics Department, Hunan Normal University, in 1999 and 2002, respectively, and the Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University, in 2006. He is currently a Professor with the School of Computer, Wuhan University, China. From 2006 to 2007, he held a postdoctoral position with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. From 2012 to 2013, he visited the University of California at Davis, Davis for one year. His main interests include computer graphics, computer vision and machine learning.



Hanqiu Sun received the M.S. degree in electrical engineering from The University of British Columbia and the Ph.D. degree in computer science from University of Alberta, Canada. She is currently an Associate Professor with the Chinese University of Hong Kong. Her research interests include virtual reality, interactive graphics/animation, real-time hypermedia, virtual surgery, mobile image/video synopsis and navigation, touch-enhanced simulations.



Feng Tang received the B.S. and M.Eng. degrees from the Computer Science Department, Zhejiang University, in 2001 and 2004, respectively, and the Ph.D. degree in computer engineering from the University of California at Santa Cruz, Santa Cruz, in 2008. Since 2009, he has been with Hewlett-Packard Laboratories as a Researcher in multimedia interaction and understanding laboratory. His research interests are in image representation, multimedia understanding, and machine learning.