Recoloring Textile Fabric Images Based on Improved Fuzzy Clustering

Zhe Zou,[1] Hui-Liang Shen,[1]* Xin Du,[1] Sijie Shao,[2] John H. Xin[2]

[1]College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, 310027, China

[2]Institute of Textiles and Clothing, the Hong Kong Polytechnic University, Hong Kong, China

Received 15 April 2015; revised 12 December 2015; accepted 12 December 2015

Abstract: This article proposes a new recoloring method for textile fabric images based on improved fuzzy local information c-means (FLICM) clustering. In the clustering algorithm, the fuzzy factor was modified so that it can produce reliable segmentation in areas with rich details. With the obtained cluster labels and pixel-wise memberships, the color of each pixel is modeled as the linear combination of the two most dominant colors. The recoloring process was then conducted by replacing the specified dominant color with user-provided target colors. Experimental results showed that the proposed method can produce natural and faithful color appearance on both printed and yarn-dyed fabric images, and outperforms the state-of-the-art. © 2016 Wiley Periodicals, Inc. Col Res Appl, 42, 115–123, 2017; Published Online 11 January 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/col.22023

Key words: coloration; recoloring; color theme design; textile fabric image; fuzzy clustering; image segmentation

INTRODUCTION

Designers often spend much time and effort in choosing proper colors to get better visual effect for products. In the textile industry, it is important but also difficult to find a pleasant and attractive color theme in fabric design, which involves much trail-and-error color dyeing.[1,2] Although the computer-aided design (CAD) tools can provide assistance, there is still a large gap between the color appearance of the virtual and physical samples. Thus it is still necessary to prepare physical samples, which is time- consuming and costly. To cope with this problem, in this article we present a novel method to recolor fabric images, with which designers can easily modify the image colors without the actual production of physical samples.

Recoloring, or colorization, is a popular image editing technique that aims to modify or change the color appearance of an image. Various approaches have been developed for the color modification of natural images. Reinhard *et al*.[3] proposed a simple and efficient method for transferring the color appearance of an image to another one by modifying the means and standard deviations of color distributions in a decorrelated color space. Levin *et al*.[4] presented an approach to coloring grayscale images by casting the colorization problem into a constrained optimization problem under the assumption that neighboring pixels with similar intensities should also have similar colors. Tai *et al*.[5,6] proposed a color transfer method based on probabilistic soft image segmentation, in which a Gaussian mixture model (GMM) and an expectation-maximization (EM) scheme are employed for spatial color smoothness. Dalmau-Cedeno *et al*.[7] introduced an interactive colorization method by using the linear combination of user-provided colors according to the probability measure field which is computed from probabilistic segmentation. Charpiat *et al*.[8] presented an automatic grayscale image colorization technique by solving a global optimization problem via multimodal predictions. Chen *et al*.[9] introduced an edit propagation algorithm for interactive image recoloring by preserving the manifold structure of all pixels in the feature space. Xu *et al*.[10] proposed a sparse control model to minimize user's work like drawing scribbles. Yoo *et al*.[11] proposed a method that transfers colors of an image to another for the local regions using their dominant colors. Jin *et al*.[12] presented

*Correspondence to: Hui-Liang Shen (e-mail: shenhl@zju.edu.cn)
Contract grant sponsor: National Natural Science Foundation of China; Contract grant numbers: 61371160 and 61271339; Contract grant sponsor: HKRITA project; Contract grant number: ITP/048/13TP.

a randomized algorithm for natural object colorization that can produce natural and rich color variations.

While the methods become more and more convenient and efficient, reference images or user-specified scribbles are still necessary inputs. In the textile industry, it is required that the user input should be minimal and the recolored fabric image should keep high color fidelity. As a fabric image always contains rich textures and numerous color regions, which is quite different from a natural scene, most of the mentioned methods could not meet this requirement. Fortunately, the number of dominant colors in a fabric image is normally not very large, and thus the recoloring process can benefit from preceding image segmentation. The recent works thus follow this way for textile image colorization. For instance, Han *et al*.[13] employed a fast multistage image segmentation algorithm to estimate the color bias filed, and used the membership function for automatic color theme design. Based on the multiphase segmentation algorithm, Zheng[14] conducted further color transfer work on fabric images. Recently, Chang *et al*.[15] introduced a novel image editing method that consists of the color palette creation and color transfer steps. The method assumes a limited number of major colors in an image and thus is applicable to fabric image recoloring. However, the methods mentioned above may result in color shifts or weaken textures, and thus the resultant images cannot keep high-fidelity color appearances.

In this article we propose a novel recoloring method for textile fabric images based on an improved fuzzy clustering algorithm. The clustering algorithm incorporates the local spatial constraint and a modification of the fuzzy factor so that it can produce better image segmentation. The color of each pixel is modeled as the linear combination of two most dominant colors, and the recoloring process can be realized by replacing the specific dominant color with user-chosen target colors. The user interaction includes specifying the number of dominant colors and selecting desired target colors, which is quite easy for practical operation. The performance of the proposed method is validated on both printed and yarn-dyed fabric images.

FUZZY CLUSTERING

Fuzzy clustering is widely used in image segmentation and color extraction.[16] Compared with hard clustering, it retains more image information by introducing the idea of partial membership. Fuzzy c-means (FCM) algorithm is one of the most popular fuzzy clustering methods, but it is sensitive to noise due to its neglect of spatial information. Various variants of FCM have been proposed,[17–20] among which, the fuzzy local information c-means (FLICM) clustering[20] is very appealing. In this section we present an improvement on FLICM so that it is more suitable to textile image segmentation.

FLICM

For an image with N pixels, the color of the ith pixel is denoted as $\mathbf{x}_i$. Suppose that the image contain c dominant colors, then in FLICM clustering, the optimal image partition is produced by minimizing the objective function

$$J_m = \sum_{i=1}^{N} \sum_{k=1}^{c} \left(u_{ki}^{m} \|\mathbf{x}_i - \mathbf{v}_k\|^2 + G_{ki} \right) \tag{1}$$

where $\mathbf{v}_k$ is the color prototype of the center of the kth cluster, $\|\cdot\|$ denotes the Euclidean norm as a distance measure. In Eq. (1), u_{ki} denotes the degree of membership of pixel i in the kth cluster, which is under the constraint

$$\sum_{k=1}^{c} u_{ki} = 1. \tag{2}$$

The fuzzification factor m determines the amount of fuzziness and is always set to 2. The fuzzy factor G_{ki} is defined as

$$G_{ki} = \sum_{j \in N_i, i \neq j} \frac{1}{1 + d_{ij}} \left(1 - u_{kj}\right)^{m} \|\mathbf{x}_j - \mathbf{v}_k\|^2 \tag{3}$$

where N_i stands for the local window around pixel i, and d_{ij} is the spatial Euclidean distance between pixels i and j.

The objective function (1) can get its local minimum by updating the membership and cluster center iteratively as

$$u_{ki} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|\mathbf{x}_i - \mathbf{v}_k\|^2 + G_{ki}}{\|\mathbf{x}_i - \mathbf{v}_j\|^2 + G_{ji}} \right)^{1/(m-1)}} \tag{4}$$

and

$$\mathbf{v}_k = \frac{\sum_{i=1}^{N} u_{ki}^{m} \mathbf{x}_i}{\sum_{i=1}^{N} u_{ki}^{m}}. \tag{5}$$

When the algorithm has converged, image segmentation can be conducted according to the maximum law of the degree of membership.

Improved FLICM

With the introduction of fuzzy factor in FLICM, the memberships of the pixels in a local window will gradually converge to a similar value in the clustering process, and thus the algorithm is robust to noise and outlier corruptions. However, the algorithm may also eliminate rich details in textile images by assigning the pixels in thin or tiny regions to the clusters of the surrounding pixels.

Some work has been carried out to improve the performance of FLICM.[21–23] In this article we propose to modify the fuzzy factor such that the algorithm can produce faithful segmentation in areas with fine details. We argue that, in the fuzzy factor, it is inadequate to consider solely the local spatial distance, but also should incorporate the color differences among the neighboring pixels as well. As a textile image seldom contains outliers, we can make an assumption that the larger the color difference between the neighboring and the current pixels, the less credible will the neighboring information be. In this regard, we quantify the credibility of the information from a neighboring pixel j to the current pixel i according to color similarity as

$$g_{ij}^{c}=\exp\left(-\frac{\|\mathbf{x}_j-\bar{\mathbf{x}}_{C_i}\|^2}{2\sigma^2}\right), \tag{6}$$

where C_i denotes the index of the cluster with which pixel i has the maximum membership,

$$C_i=\arg_k\{\max\{u_{ki}\}\}, k=1,2,\ldots,c. \tag{7}$$

In Eq. (6), $\bar{\mathbf{x}}_{C_i}$ is the mean color of the pixels currently in cluster C_i, and σ controls the level of color similarity. In this article, we set σ as the standard deviation of the pixels currently in cluster C_i so that the color similarity term g_{ij}^{c} can adapt to different clusters.

By writing the spatial constraint as

$$g_{ij}^{d}=\frac{1}{1+d_{ij}}, \tag{8}$$

and the local pixel distribution as

$$g_{kj}=(1-u_{kj})^m\|\mathbf{x}_j-\mathbf{v}_k\|^2, \tag{9}$$

the new fuzzy factor is defined as

$$\tilde{G}_{ki}=\sum_{j\in N_i, i\neq j} g_{ij}^{d}g_{ij}^{c}g_{kj}. \tag{10}$$

With the introduction of this new fuzzy factor, the clustering algorithm has the objective function

$$J_m=\sum_{i=1}^{N}\sum_{k=1}^{c}\left(u_{ki}^{m}\|\mathbf{x}_i-\mathbf{v}_k\|^2+\tilde{G}_{ki}\right). \tag{11}$$

Similar to the original FLICM algorithm, the membership is iteratively updated as

$$u_{ki}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\|\mathbf{x}_i-\mathbf{v}_k\|^2+\tilde{G}_{ki}}{\|\mathbf{x}_i-\mathbf{v}_j\|^2+\tilde{G}_{ji}}\right)^{1/(m-1)}}, \tag{12}$$

and the color of the cluster center is updated as

$$\mathbf{v}_k=\frac{\sum_{i=1}^{N}u_{ki}^{m}\mathbf{x}_i}{\sum_{i=1}^{N}u_{ki}^{m}}. \tag{13}$$

The algorithm of the improved FLICM clustering is outlined as follows:

Step 1. Set the number of clusters c, fuzzification m, and stopping tolerance ε.

Step 2. Initialize the membership matrix $U^{(0)}$.

Step 3. Set the loop counter $t=0$.

Step 4. Compute the color of the cluster center $\mathbf{v}_k$ according to Eq. (13).

Step 5. Update the membership $U^{(t)}$ according to Eq. (12).

Step 6. Stop if $\max\{U^{(t)}-U^{(t-1)}\} < \varepsilon$; otherwise set $t=t+1$ and go to Step 4.

Image Segmentation

In this article the fuzzy clustering is applied on the textile fabric images using the CIELAB color space, and the Euclidean distance in Eq. (6) is actually the CIELAB color difference. After clustering, each pixel is assigned with the cluster label that corresponds to the maximum membership. Figure 1 shows the image segmentation maps and corresponding resultant recolored images based on the original and improved FLICM algorithms. For demonstration purpose, the segmentation maps are displayed in pseudo-colors. It is observed from the close-up views that, for the original FLICM algorithm, some pixels with green colors are mislabeled because its fuzzy factor considers only spatial constraint. In comparison, the improved FLICM algorithm produces a better segmentation map due to the introduction of color similarity term in the fuzzy factor. Correspondingly, colors of these pixels are less

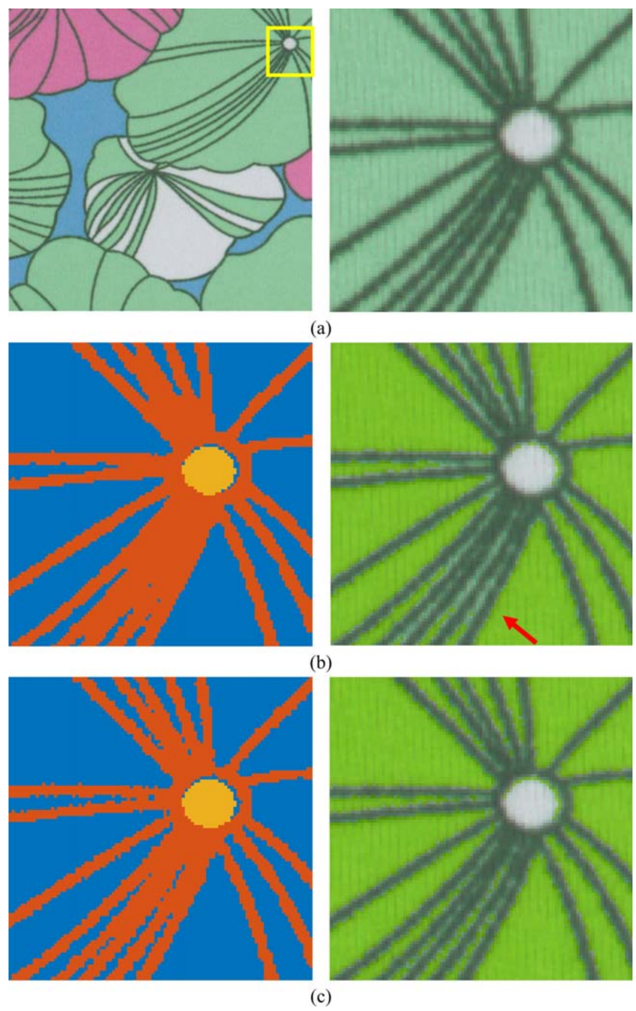

Fig. 1. An example of fabric image segmentation and recoloring. (**a**) Original printed fabric image and a close-up view. (**b**) Segmentation map by the original FLICM algorithm and resultant recolored image. (**c**) Segmentation map by the improved FLICM algorithm (proposed) and resultant recolored image. The segmentation maps are displayed in pseudo colors. In (b), color shift is indicated by the *red arrow*.

biased in the recolored images when employing the improved FLICM algorithm in image segmentation.

RECOLORING

Color Modeling

In the recent works,[13,14] the color of a pixel is represented by the linear combination of all dominant colors, while the weights are simply the memberships obtained from the clustering process. The major problem of this treatment is that, as the memberships do not exactly reflect the weighting of dominant colors, the recolored image usually suffers from color artifacts.

In this article, we model the color of a pixel as the linear combination of at most two dominant colors. This is reasonable for textile images, as the solid-color regions have only one color, while the colors of pixels in transitional areas are usually affected by two adjacent regions.

A spectral reflectance value $r(\lambda)$ of a pixel, where λ denotes the wavelength, can be decomposed into the linear combination of the reflectances of two dominant colors, $r_1(\lambda)$ and $r_2(\lambda)$, as

$$r(\lambda)=\alpha r_1(\lambda)+\beta r_2(\lambda)+e(\lambda), \tag{14}$$

where $e(\lambda)$ denotes the approximation error in the reflectance space. The tristimulus values of a reflectance $r(\lambda)$ are computed as

$$\begin{aligned} X&=K\cdot\sum_{\lambda}\bar{x}(\lambda)L(\lambda)r(\lambda)\Delta\lambda \\ Y&=K\cdot\sum_{\lambda}\bar{y}(\lambda)L(\lambda)r(\lambda)\Delta\lambda \\ Z&=K\cdot\sum_{\lambda}\bar{z}(\lambda)L(\lambda)r(\lambda)\Delta\lambda, \end{aligned} \tag{15}$$

where

$$K=\frac{100}{\sum_{\lambda}\bar{y}(\lambda)S(\lambda)\Delta\lambda}, \tag{16}$$

$\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ denote the CIE color matching functions, $L(\lambda)$ represents the spectral power distribution of the illuminant, and $\Delta\lambda$ denotes the wavelength interval. Then Eq. (14) can be reformulated in the CIEXYZ color space as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}=\alpha\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}+\beta\begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix}+\mathbf{e}, \tag{17}$$

where $(X_1,Y_1,Z_1)^{\mathrm{T}}$ and $(X_2,Y_2,Z_2)^{\mathrm{T}}$ are, respectively, the corresponding tristimulus values of reflectance $r_1(\lambda)$ and $r_2(\lambda)$, and the 3×1 vector $\mathbf{e}$ denotes the residual error.

According to Eqs. (14) and (17), the recoloring process can be applied on either multispectral images or color images. Considering that the aim of recoloring in the textile industry is to assist color scheme design, in the following we conduct recoloring in CIEXYZ color space. The extension to reflectance space, as shown, is straightforward.

Recoloring Algorithm

In the following, we consider the recoloring process of pixel i whose CIEXYZ values are denoted by a 3×1 vector $\mathbf{x}_{\mathrm{i}}$. For the sake of notation simplicity, the color is denoted as $\mathbf{x}$ by dropping the subscript. Suppose that the textile image has been segmented by the proposed fuzzy clustering method, and the dominant colors, as well as the memberships of the pixels, have been obtained. We also suppose that, for pixel i, the cluster c_1 has the maximum membership, while the cluster c_2 has the second maximum membership. Then the mean colors of the clusters with c_1 and c_2, which are denoted as $\mathbf{v}_1$ and $\mathbf{v}_2$, are the two most dominant colors for the pixel. According to Eq. (17), the color $\mathbf{x}$ of the pixel is model as

$$\mathbf{x}=\alpha\mathbf{v}_1+\beta\mathbf{v}_2+\mathbf{e}, \tag{18}$$

where α and β are the coefficients corresponding to the dominant colors $\mathbf{v}_1$ and $\mathbf{v}_2$, and the vector $\mathbf{e}$ denotes the residual error.

In Eq. (18), the optimal coefficients α and β should minimize the error term $||\mathbf{e}||^2$. In addition, as the color $\mathbf{x}$ is the mixture of the two dominant colors, it is reasonable to require that the coefficients should be nonnegative. Then the solution of coefficients is formulated as

$$\begin{aligned} \{\alpha,\beta\} &= \arg\min{}_{\alpha,\beta}||\mathbf{x}-\alpha\mathbf{v}_1-\beta\mathbf{v}_2||^2 \\ &\text{s. t.}\quad \alpha\geq0,\quad \beta\geq0, \end{aligned} \tag{19}$$

which is a quadratic programming problem. As the direct employment of quadratic programming is computationally intensive, in the following we first solve the coefficients under the least-squares criterion and then enforce the coefficient to be nonnegative.

Let $\mathbf{V}=(\mathbf{v}_1,\mathbf{v}_2)$, then the coefficient $\mathbf{c}=(\alpha,\beta)^{\mathrm{T}}$ can be solved using least-squares,

$$\mathbf{c}=(\mathbf{V}^{\mathrm{T}}\mathbf{V})^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{x}. \tag{20}$$

Accordingly, the residual error is computed as $\mathbf{e}=\mathbf{x}-\mathbf{Vc}$. In case that one coefficient, for example β, is negative, we simplify the mixture model Eq. (18) to

$$\mathbf{x}=\alpha\mathbf{v}_1+\mathbf{e}, \tag{21}$$

which implies that $\beta=0$. Then α is solved as

$$\alpha=(\mathbf{v}_1^{\mathrm{T}}\mathbf{v})^{-1}\mathbf{v}^{\mathrm{T}}\mathbf{x}. \tag{22}$$

As the dominant color $\mathbf{v}_1$ is close to $\mathbf{x}$, we can ensure that $\alpha>0$. Accordingly, the residual error is $\mathbf{e}=\mathbf{x}-\alpha\mathbf{v}_1$.

The pixel can be recolored according to the color mixture model based on the coefficients α and β computed in Eq. (20) or (22). Let $\mathbf{u}$ be the user-chosen target color, then the new color of the pixel can be computed by replacing the dominant color $\mathbf{v}_1$ with $\mathbf{u}$ in the linear model,

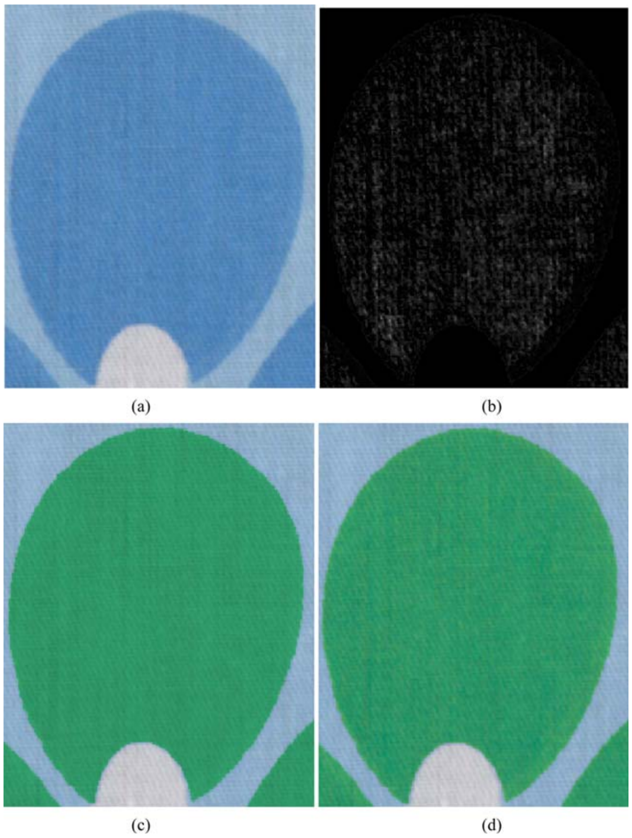

Fig. 2. Recoloring results with and without incorporating the residual error term. (**a**) Original image. (**b**) Residual error, enhanced for visualization. (**c**) Recoloring result without the residual error term. (**d**) Recoloring result with the residual error term.

$$\hat{\mathbf{x}} = \alpha\mathbf{u} + \beta\mathbf{v}_2. \tag{23}$$

Note that Eq. (23) represents an ideal recoloring process that ignores the residual error $\mathbf{e}$. As the residual error also contains partial texture information, the recoloring results produced by Eq. (23) may appear not very realistic. In this regard, we employ the following model for recoloring,

$$\hat{\mathbf{x}} = \alpha\mathbf{v}_0 + \beta\mathbf{v}_2 + \eta\mathbf{e}, \tag{24}$$

where the parameter η compensates the texture effect. In this article we set η=0.5. Figure 2 shows the recoloring results with and without incorporating the residual error term. It is observed that the color appearance of the resultant image seems more natural when considering the residual error term.

Recoloring Process

In this article, the modification of color scheme requires minimum user interaction. First, the user provides the number of dominant colors and segments the image via the improved fuzzy cluster algorithm. Then, the user edits the image by replacing the specific dominant color with a target color. When two or more colors are to be replaced, we can simply iterate the recoloring algorithm with different target colors.

As mentioned, the color of a pixel can be modeled using Eq. (18) or (21), depending on whether the coefficients α and β are nonnegative. Our investigation indicates that the model Eq. (18) is necessary to the pixels in the transitional area between two color regions, while model Eq. (21) can describe most pixels inside a region. Nevertheless, as coefficients are automatically determined by the recoloring algorithm, it is not necessary for the user to choose the proper model.

EXPERIMENTS

In the experiment, we acquired printed and yarn-dyed fabric images using a multispectral imaging system.[24] The CIEXYZ and CIELAB format images are obtained from the multispectral images under the CIE standard illuminant D65. The improved fuzzy clustering is applied on the CIELAB image, while the recoloring is applied on the CIEXYZ image. The recolored images are finally transformed to sRGB color space[25] for visualization.

We evaluated the proposed method, as well as the methods by Han *et al.*,[13] Zheng,[14] Chen *et al.*,[9] and Chang *et al.*,[15] on a number of fabric images. Han's method, Zheng's method, Chen's method and the proposed method are all implemented in MATLAB 2010. Chang's method was implemented by its authors in Javascript with a graphic user interface (GUI). We modified the code such that, despite the specified regions, the luminance values of other regions would not change in the recoloring process. All these methods run on a personal computer with 2.13 GHz CPU (Intel Xeon E5506) and 12 GB RAM.

Note that the propose method, as well as Han's, Zheng's, and Chang's methods, are all based on color clustering. For these methods, the number of clusters, c, in an image is manually specified. In Chen's method, scribbles are manually supplied to indicate which colors are to be changed and which colors should be kept unchanged.

Comparison

Figure 3 shows the recoloring results of a printed fabric image, in which two colors of the cherry pattern are to be modified. It can be seen that the proposed method produces a natural appearance in the recolored image. In the image produced by Han's method, the background color appears yellowish though we do not intend to change it. In addition, both Han's and Zheng's methods produce unpleasant color speckles on the cheery pattern. The reason is that these two methods simply adopt the memberships in the color mixture models, which fail to guide the recoloring process. It is noted that a large number of scribbles have been provided as user input in Chen's method. However, as illustrated in the highlighted area, the resultant image is still not satisfactory. For Chang's method, its resultant image produces wrong color at the left-bottom due to incorrect image segmentation.

Figure 4 shows the recoloring results of a yarn-dyed fabric, in which the red color is to be modified to purple.

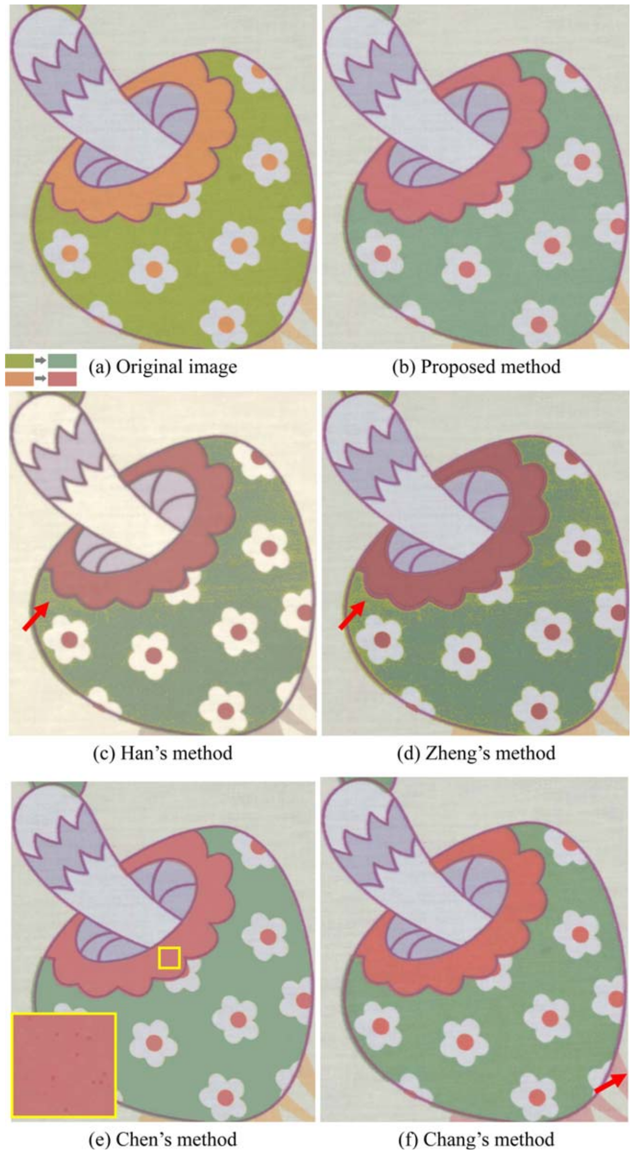

Fig. 3. Recoloring results of a printed fabric image. The two source-target color pairs are shown under the original image. The flaws are highlighted by *red arrows* or *yellow box*.

For Han's method, the resultant image appears blurry in the white/dark area, and color transition between the purple yarn and dark background seems too abrupt. The abrupt color transition also exists in the image produced by Zheng's method. In the result by Chen's method the small regions get connected to each other. Chang's method produces good visual effect, but color shift can be perceived when comparing the recolored yarns with respect to the target color. In comparison, the proposed method produces natural appearance and does not introduce color shift.

Figures 5 and 6 show additional recoloring results on printed fabric images. In Fig. 5, all pattern colors, except the white background, are to be changed. The proposed method produces an image with natural color appearance. In comparison, the textures are clearly weakened in the recolored images produced by Zheng's and Chen's methods. Also, Chang's method introduces noticeable color bias on the white background. In Fig. 6, the recoloring task in is difficult as the region boundaries are no clear in

Fig. 4. Recoloring results of a printed fabric image.

Fig. 5. Recoloring results of a printed fabric image. Close-up views are shown for detail comparison.

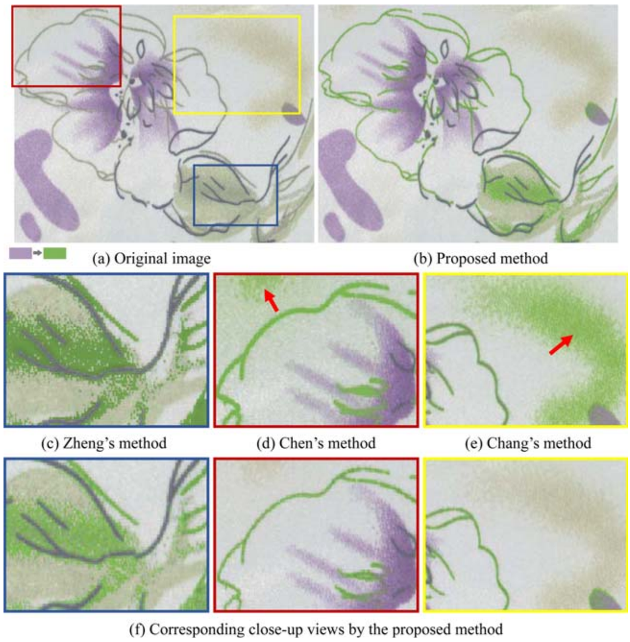

Fig. 6. Recoloring results of a printed fabric image. Close-up views are shown for detail comparison.

this printed fabric image. Nevertheless, the proposed method still performs quite well in recoloring the long curve and simultaneously keeps other colors unaltered. In comparison, Zheng's method yields an image with unnatural color appearance, and Chen's and Chang's methods lead to unexpected color changes as indicated by red arrows.

Figures 7 and 8 show two more recoloring results on yarn-dyed fabric images. As observed in Fig. 7, the recolored yarns appear unnaturally flat in the resultant image by Han's and Zheng's methods. The result by Chen's method contains obvious artifacts. The proposed method and Chang's method both produce satisfactory recolored images. In Fig. 8, the proposed method again yields good recoloring results. In comparison, the resultant images by Zheng's and Chen's methods are not acceptable, while

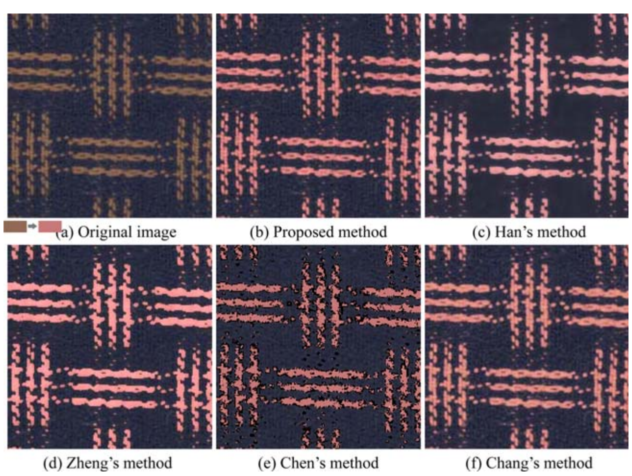

Fig. 7. Recoloring results of a yarn-dyed fabric image.

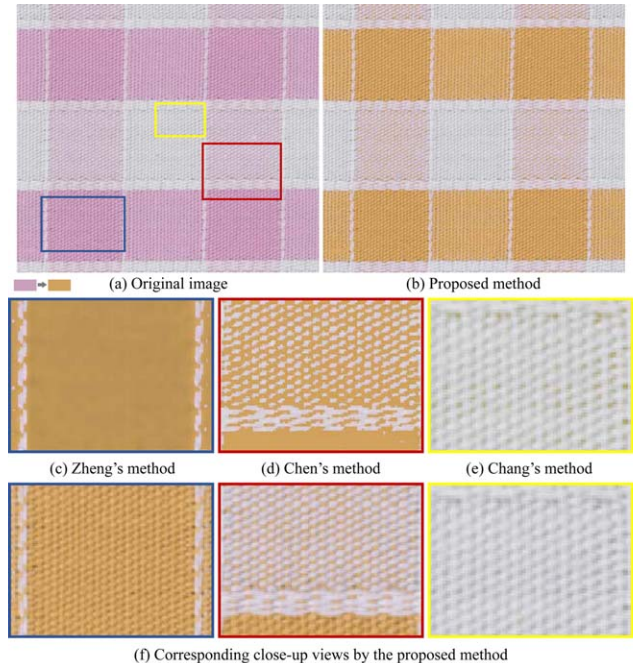

Fig. 8. Recoloring results of a yarn-dyed fabric image. Close-up views are shown for detail comparison.

that by Chang's method contains perceivable color changes in shade areas.

In additional to recoloring effect, it is also of interest to evaluate the running efficiency of the recoloring methods. The computational cost of the proposed method is mainly determined by the clustering process which is of complexity $O(Nc)$. Chang's method is implemented in Javascript and should be efficient as it does not consider spatial information in the clustering process. For Han's and Zheng's methods, the estimation of color bias field is computationally intensive. In Chen's method, locally linear embedding (LLE) and sparse-system based recoloring both requires heavy computation. Table I lists the running time of these recoloring methods obtained on an image with 500×450 pixels. The computational cost of the proposed method is about 43 seconds, indicating it runs faster than all the competitors except Chang's method. It is worth noting that, when implemented in C^{++}, the running time of the proposed method is about 3 seconds, which is sufficient for interactive recoloring tasks.

TABLE I. Running time (unit: seconds) of the recoloring methods obtained on an image with 500 × 450 pixels.

Method	Platform	Clustering/LLE	Recoloring	Total
Han[13]	Matlab	89.6	0.3	89.9
Zheng[14]	Matlab	89.6	0.5	90.1
Chen[9]	Matlab	136.2	45.1	181.3
Chang[15]	Javascript	5	4.9	9.9
Proposed	Matlab	40.3	2.9	43.2

Note that Chen's method is based on locally linear embedding (LLE) and other methods are based on clustering.

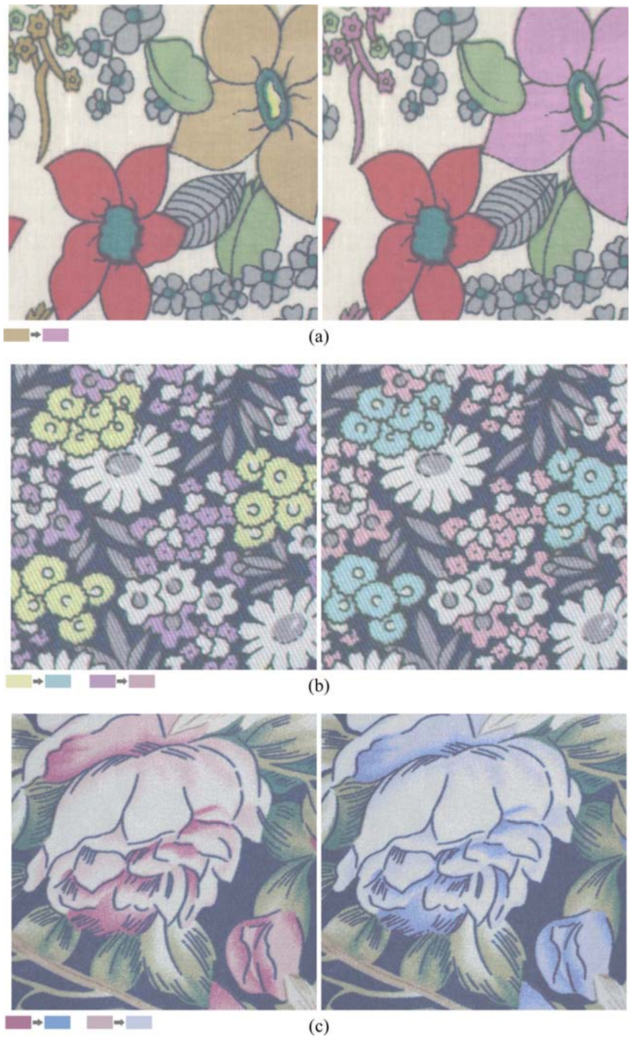

Fig. 9. Recoloring results of printed fabric images using the proposed method. Left column: original images; right column: results.

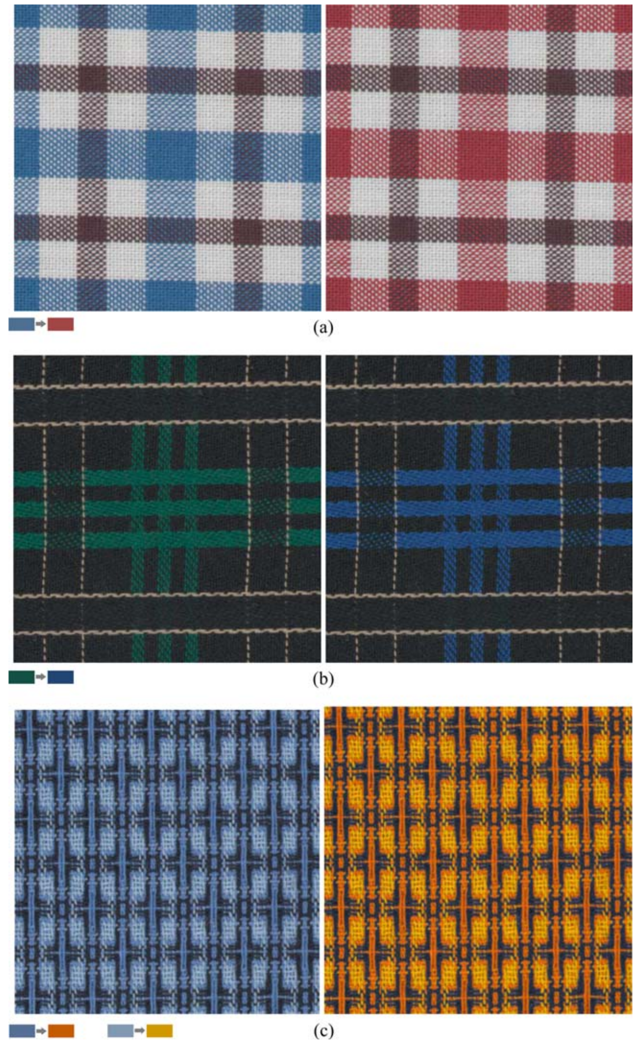

Fig. 10. Recoloring results of yarn-dyed fabric images using the proposed method. Left column: original images; right column: results.

More Results

Figures 9 and 10 show more recoloring results of the proposed method on printed and yarn-dyed fabric images, respectively. In Fig. 9(a) one color is replaced while in both Figs. 9(b) and 9(c) two colors are replaced. It is observed that the texture effect is well preserved in Fig. 9(b). Also, the color appearance of the resultant image in Fig. 9(c) is very natural. Figure 10 shows the recoloring results on three yarn-dyed fabric images. It can be seen that the color appearances are quite realistic and the texture effects are kept well.

CONCLUSIONS

In this article we have presented a new recoloring method for textile fabric images, which consists a fuzzy clustering algorithm and a recoloring algorithm. The fuzzy clustering is improved by introducing a color similarity measure so that it can deal with fine details in textile fabric images. In the recoloring algorithm, the color of a pixel is modeled as the linear combination of two most dominant colors, with nonnegative constraint on the coefficients. The color scheme can then be modified by replacing the specific dominant color with target colors. Experimental results validate the performance of the proposed method on both printed and yarn-dyed fabric images.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their insightful comments.

1. Sauvaget C, Manuel S, Vittaut J-N, Suarez J, Boyer V. Segmented images colorization using harmony. In Proceedings of the International Conference on Signal-Image Technology and Internet Based Systems, Kuala Lumpur, Malaysia, 2010. p 153–160.
2. Lin S, Ritchie D, Fisher M, Hanrahan P. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. ACM Trans Graph 2013;32:37
3. Reinhard E, Ashikhmin M, Gooch B Shirley P., Color transfer between images. IEEE Comput Graph Appl 2001;21:34–41.

4. Levin A, Lischinski D, Weiss Y. Colorization using optimization. ACM Trans Graph 2004;23:689–694.
5. Tai Y-W, Jia J, Tang C-K. Local color transfer via probabilistic segmentation by expectation-maximization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005. p 747–754.
6. Tai Y-W, Jia J, Tang C-K. Soft color segmentation and its applications. IEEE Trans Pattern Anal Mach Intell 2007;29:1520–1537.
7. Dalmau-Cedeno O, Rivera M, Mayorga PP. Computing the α-channel with probabilistic segmentation for image colorization. In Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007. p 1–7.
8. Charpiat G, Hofmann M, Schölkopf B. Automatic image colorization via multimodal predictions. In Proceedings of the European Conference on Computer Vision, Marseille, France, 2008. p 126–139.
9. Chen X, Zou D, Zhao Q, Tan P. Manifold preserving edit propagation. ACM Trans Graph 2012;31:132.
10. Xu L, Yan Q, Jia J. A sparse control model for image and video editing. ACM Trans Graph 2013;32:197.
11. Yoo J-D, Park M-K, Cho J-H, Lee KH. Local color transfer between images using dominant colors. J Electron Imag 2013;22:033003.
12. Jin S-Y, Choi H-J, Tai Y-W. A randomized algorithm for natural object colorization. Comput Graph Forum 2014;33:205–214.
13. Han Y, Zheng DJ, Baciu G, Feng XC, Li M. Fuzzy region competition-based auto-color-theme design for textile images. Text Res J 2013;83:638–650.
14. Zheng D. A novel method for fabric color transfer. Color Res Appl 2015;40:304–310.
15. Chang H, Fried O, Liu YM, DiVerdi S, Finkelstein A. Palette-based photo recoloring. ACM Trans Graph 2015;34:139.
16. Vantaram SR, Saber E. Survey of contemporary trends in color image segmentation. J Electron Imag 2012;21:040901.
17. Ahmed MN, Yamany SM, Mohamed N, Farag AA, Moriarty T. A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data. IEEE Trans Med Imag 2002;21:193–199.
18. Chen S, Zhang D. Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Trans Syst Man Cybern Part B: Cybern 2004;34:1907–1916.
19. Cai W, Chen S, Zhang D. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. Pattern Recogn 2007;40:825–838.
20. Krinidis S, Chatzis V. A robust fuzzy local information c-means clustering algorithm. IEEE Trans Image Process 2010;19:1328–1337.
21. Gong M, Liang Y, Shi J, Ma W, Ma J. Fuzzy c-means clustering with local information and kernel metric for image segmentation. IEEE Trans Image Process 2013;22:573–584.
22. Li N, Huo H, Zhao YM, Chen X, Fang T. A spatial clustering method with edge weighting for image segmentation. IEEE Geosci Remote Sens Lett 2013;10:1124–1128.
23. Xiang D, Tang T, Hu C, Li Y, Su Y. A kernel clustering algorithm with fuzzy factor: Application to SAR image segmentation. IEEE Geosci Remote Sens Lett 2014;11:1290–1294.
24. Shen HL, Yao JF, Li C, Du X, Shao SJ, Xin JH. Channel selection for multispectral color imaging system using binary differential evolution. Appl Opt 2014;53:634–642.
25. Stockes M, Anderson M, Chandrasekar S, Motta R. A standard default color space for the internet – sRGB, 1996. http://www.w3.org/graphics/color/srgb.html

Published Online 21 October 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/col.22023

Erratum

Calibrating Low-Scattering Samples Using Kubelka-Munk Model

Michael H. Brill

Datacolor, 5 Princess Road, Lawrenceville, New Jersey 08648

In the subject article,[1] Eq. (12) was incorrectly typeset to include $R_g + u$ in the argument of the exponential. The correct rendition of Eq. (12) is as follows:

$$v=\left(cR_g-c-1-d\right)\frac{\exp\left(2dKX\right)}{\left(R_g+u\right)}. \tag{12}$$

I thank Frank Ligterink for bringing this error to my attention.

1. Brill MH, Li YQ. Calibrating low-scattering samples using Kubelka-Munk model. Color Res Appl 2016; 41:399–401. DOI: 10.1002/col.21965

Published Online 21 October 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/col.22096