Translucent Image Recoloring through Homography Estimation

Yifei Huang, Changbo Wang and Chenhui Li[†]

School of Computer Science and Software Engineering, East China Normal University, China

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

Figure 1: *Comparisons of the cup recoloring. (a) The source image. (b) The target color palette. (c) The result from Chang et al. [CFL*15]. (d) The result from Han et al. [HXB*17]. (e) Our method through homography estimation. Our method retains more detail than other methods, such as the petal in the cup.*

Abstract

Image color editing techniques are of great significance for users who wish to adjust the image color. However, previous works paid less attention to the translucent images. In this paper, we propose a new method to recolor the translucent images while preserving detailed information and color relationships of the source image. We consider the recolor problem as a location transformation problem and solve it in two steps: automatic palette extraction and homography estimation. First, we propose the Hmeans method to extract the dominant colors of the source image based on histogram statistics and clustering. Then, we propose homography estimation to map the source colors to desired colors in the CIE-LAB color space. Further, we adopt a non-linear optimization approach to improve the result generated by the last step. The proposed method maintains high fidelity of the source image. Experiments have shown that our method generates a state-of-the-art visual result, in particular in the shadow areas. The source images with ground truth generated by a ray tracer further verify the effectiveness of our method.

CCS Concepts

•***Computer Graphics*** $\rightarrow$ ***Image Editing;***

1. Introduction

Color plays an important role in human feelings about the world. Different colors will result in different feelings. For example, the color red will attract people's attention while increasing breathing rates [NE04]. It is of great importance for users to recolor images to their desired style. In addition, color transfer has been well-studied in image editing. During color transfer, the color relationships and source image details are preserved. However, previous work paid less attention to translucent images. Translucent images such as cup, gem, water and so on are very common in our daily life. It is often labor intensive to recolor translucent images due to the fact that the translucent consists of semitransparent objects. The techniques to recolor translucent images can help us save a lof of labor. In particular, it will avoid manual selection of the transparent areas during the process of image editing.

The main difficulties in recoloring translucent images are as follows. Compared to opaque image, there are more detailed information that needs to be preserved during the recoloring process, caused by by multiple reflections of light and semitransparent objects. In particular, the shadow area of the translucent image is difficult to handle. On the one hand, the correspondence between the shadow area and its object should be considered during the recoloring. On the other hand, it is troublesome to segment the shadow area based on either color decomposition or image segmentation algorithms. Originally, the color transfer approach was based on an example image. Reinhard et al. [RAGS01] applied the statistical color characteristics of an example image to the source image. However, the simple image features computed by the mean and standard deviation lacked fidelity to the image details of the source image. Then, Xiao et al. [XM09] presented the gradient-preserving method to resolve this problem. They took both the histogram and the gradient information of the source image into consideration. All of the methods above require an example image. However, the example image, which adjusts the source image to the users' preferred image, can often not be obtained. In addition, the researcher adopted the text to describe the preferred style. Liu et al. [LCUR14] apply the preferred style obtained from a keyword such as "Spring" or "Sepia" to the source image. Nevertheless, it is not convenient for users to describe their desired style in the text. Recently, palette-based color transfer methods have given us a new direction. The palette-based color transfer methods allow users to adjust the image color to any color they want. The tool designed by Chang et al. [CFL*15] is easy to use and generates state-of-the-art results. They allowed users to edit the color directly according to the color palette and used the RBF function to map the source colors to target colors. Zhang et al. [ZXST17] consider the color mapping from source colors to target colors as a color decomposition optimization while preserving the inherent color characteristics. In spite of the remarkable results, the methods above lack fidelity to the source image when the source image is translucent. Furthermore, it is worth noting that $\mathscr{L}0$ gradient-preserving method proposed in [WZL*17] can recolor images with either an example target image or a target palette. The authors of [WZL*17] stated the shadow areas of some images are not easily handled. The main difference between [WZL*17] and our work is that we focus on a new problem which is to recolor translucent images.

This paper introduces a new palette-based method to recolor translucent images while maintaining high fidelity to the source image. Our method can be divided into two steps: automatic palette extraction and homography estimation. First, we extract the dominant colors from the source image to form a representable palette. The extracted dominant colors can not only be used to represent the colors of the source image but also involve the color relationships in the color space. We compute the homography matrix to transform the source colors to the target colors, motivated by the image alignment [Sze06]. We consider the color point in the color space as the pixel in one image. Second, we employ image transformation techniques to transfer the color. To be clear, all operations are performed in the CIE-LAB color space to provide the perceptual result. The native transformation would cause unsatisfactory results. Some pixels would go out of gamut during the transformation. Therefore, we non-linearly renormalize the colors to the visible area in the color space.

In sum, the main contributions of this paper are as follows:

- Propose a new method to recolor translucent images which is a new problem in image recoloring research field and therefore fill the gap.
- Describe a novel experimental method by generating ground truth images from a ray tracer to evaluate different recoloring methods.

2. Related Work

Our work on recoloring the translucent images is based on four streams of prior research: the example-image based color transfer, edit propagation, palette based recoloring, and image alignment.

2.1. The Example-image based Color Transfer

Example-based color transfer applies the example image color style to the source image. For instance, given an example image, its style is transferred to the source image. Reinhard et al. [RAGS01] apply the mean and standard deviation information from the target image to the source image. When the example image differs from the source image greatly, the algorithm requires users to separate the swatches. The color relationships in the local area are considered in the work of Tai et al. [TJT05]. They use a modified EM method to recolor the source image seamlessly. The colors are divided into different categories based on their psychophysical experiments in [CSUN06] and the pixels in the same categories were mapped. However, the aforementioned methods generates images with low fidelity to the source image details. The gradient-preserved method as shown in [XM09] is proposed to solve this problem. The dense correspondences from the source image to the target image were utilized to transfer the style in [HSGL11]. A new histogram reshaping method is developed in [PR11] to achieve creative effects by manipulating the differently scaled histograms. The source image is segmented into different regions based on dominant colors [YPCL13], and they match the target color to different regions. More recently, Wang et al. [WZL*17] further propose the $\mathscr{L}0$ gradient-preserving method to preserve the detail information of the source image. With the rapid development of machine learning, Luan et al. [LPSB17] introduce a new photorealism regularization term to transfer the style, especially color from the reference image to the source image, through neural networks.

It is a remarkable fact that the example-image-based methods above require an example or reference image. However, it is not convenient to obtain the desired example image for the user. Although the key texts are used to find the example images [LCUR14] and the style of the image collections is applied to the source image, it is still not easy to describe the style that the user wishes in the special cases.

2.2. Edit Propagation

The edit propagation transfers the users' input color to the image. For instance, the user first draws colored strokes on the image and the users' edits are propagated to recolor the whole image. Levin et al. [LLW04] consider coloring the gray image with some colored strokes as an optimization problem. They assume that the neighboring pixels that have the same intensities should have the same colors. How to propagate the local tonal adjustment to the entire image is addressed in [LFUS06]. However, they focus on the adjustment of tone, not color. Qu et al. [QWH06] focus on colorizing the black-and-white manga with the colored strokes. For propagating the initial adjustment of the image, An et al. [AP08] use energy optimization to obtain the final edit. Energy optimization is accelerated by the k-d tree while using adaptive clustering in [XLJ*09]. For editing the image in real time, Bie et al. [BHW11] propose a

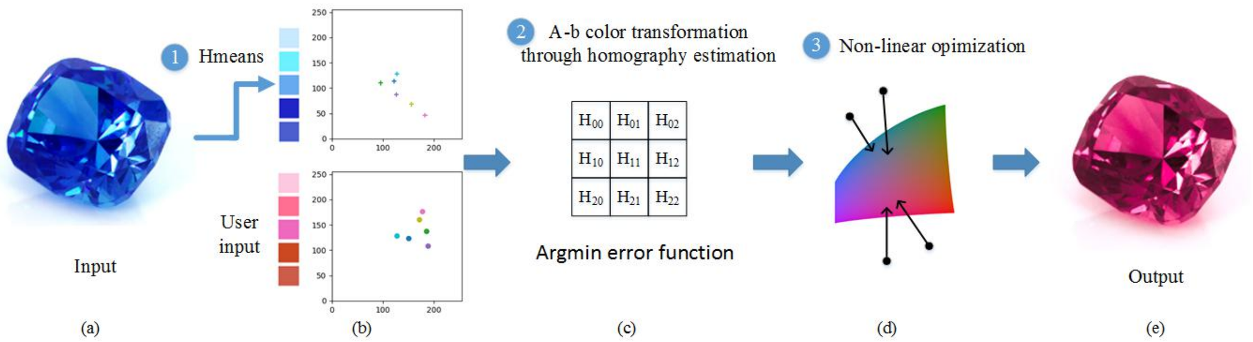

Figure 2: *The flowchart of our method. (a) The input image. (b) The extracted dominant colors of the input image by Hmeans and the target colors are in the top and bottom correspondingly. The right column displays colors locations in the a-b plane. (c) A-b color transformation through homography estimation in the a-b plane. (d) Non-linear optimization to correct the pixels that are out-of-gamut. (e) The output image after color transformation.*

two-step edit method. They first solve edits at reduced clusters and then interpolate the cluster edits to the individual pixels. Moreover, a sparse control model is promised in [XYJ13] to reduce user interaction. For editing high-resolution images, Chen et al. [CZL*14] only compute the representable samples obtained by sparse dictionary learning rather than the whole image, which reduces time and memory consumption. Recoloring time-lapse videos through interactive color retargeting is proposed in [LDLM15]. More recently, Zhang et al. [ZZI*17] map a grayscale image to an output colorization through several priors which are learned from a deep neural network.

The aforementioned methods are considered as optimization. Instead, Li et al. [LJH10] re-formulate the edit Propagation as an RBF interpolation. Xiao et al. [XY*11] further accelerate this method through the quadtree-based data structure. The stroke-based input can also be the input of our method. However, it is easy to misunderstand the users' intention for translucent images. For example, if you draw a stoke in the cup tea in Fig. 1, it is difficult to know you wish to recolor the cup or the tea.

2.3. Palette-based Color Transfer

Palette-based color transfer applies the target color palette to the source image. When the initial palette of the source image is computed, the user can edit the palette to the final palette. The palette based methods solve the change in all colors when some of the target colors are specified by the user. A lot of possible recolored images are recommended for users to navigate and choose through a Gaussian Mixture Model (GMM) [SSCO09]. On the other hand, Wang et al. [WYW*10] first segment the image and recolor the image with a pre-defined color theme. They focused on how to assign a better palette color to the image. A probabilistic factor graph is trained on a lot of human artists' work for coloring the 2D pattern templates [LRFH13]. They focus on the uncolored pattern templates. The association between the initial palette colors and the final palette colors is addressed in [CFL*15], and the deployed online tool is easy to use. They first select the palette automatically through a variant of K-means algorithm and convert the colors in L and *a-b* channel separately in the CIE-LAB color space. Further, for mapping the target colors to the source image, different constraints such as color fidelity, smoothness, and regularization are minimized in [ZXST17] while preserving inherent color characteristics. They consider the mapping from the source color palette to the target color palette as a color decomposition optimization. In the same period, Han et al. [HXB*17] also employ the decomposition process for recommending fabric images with different color styles. They further improve the performance of the decomposition through total generalized variation.

Our work also follows this direction. However, we focus on how to recolor the translucent images, in particular the shadow area of the source image. To the best of our knowledge, no previous work has paid attention to the translucent image.

2.4. Image Alignment

Image alignment is the process of automatically aligning two images that have common areas. Feature-based alignment is one of the most popular approaches to aligning two images automatically. There are a lot of feature detection methods such as Harris corner detection [HS88], scale-invariant feature detection [Low04], and fast robust feature detection [BTVG06]. The informative features can be used to match the object accurately. After detecting the features of the images, techniques such as hashing, nearest neighbors, and exhaustive search can be used to match the features. The pixel correspondences are calculated between two images before further image alignment. The transformation matrix is calculated to locate the pixels of the image in a process of image alignment. Related prior works include RANSAC [FB87] and FLANN [ML09].

Our work is motivated by the image alignment between two images. The color point in the color space is similar to the pixel in the image. Similarly, the color transformation in the color space is similar to the pixel transformation in the image.

3. Method

This section presents a novel, straightforward method for recoloring the image, especially the translucent image. The flowchart of our method is shown in Fig. 2. As we are motivated by the image alignment, we consider the color transformation in the color space as the pixel coordinate transformation in the Cartesian coordinate system. Since key points in the image alignment are used to constitute the topology of the image, we propose Hmeans to extract dominant colors, which will be introduced in Section 3.1. The dominant colors are the same as the key points, which are used to represent the color of the source image in the *a-b* plane.

After extracting the dominant colors, we need to map the source colors to the target colors that user prefers. For mapping the source colors to the target colors, we propose using homography estimation that is also used in image alignment to transfer key points, as introduced in Section 3.2.

Naive transformation will cause some pixels to be out-of-gamut. Therefore, a non-linear optimization approach is proposed to push the out-of-gamut pixels to the visible area, as introduced in Section 3.3. In addition, we discuss the influence of the number of dominant colors in Section 3.4.

3.1. Hmeans

This section presents how to automatically extract dominant colors from the source image based on histogram statistics and variant cluster methods of K-means [Llo82]. We name our method Hmeans. Our intent is to extract enough k dominant colors $\{C_i\}$, which not only form a palette of the source image but also represent the color relationships in the color space. You can think of these dominant colors as the key points in the image. The more dominant colors we extract, the more accurate the transformation will be. However, it will be very difficult for users to assign the target colors if there are too many colors. We must create a balance between the number of dominant colors and user convenience.

There are many approaches to extracting the dominant colors in previous works. For example, O'Donovan et al. [OAH11] utilize the rated colors to extract the dominant colors from an image based on color compatibility. The five dominant colors palette is built after studying how people use color themes from a number of natural images [LH13]. A weighted (k+1)-means method is used in [CFL*15], while the k is defined by the user. However, none of their approaches conform to our requirements because they pay less attention to the color relationship representation in the color space.

If we naively use K-means to cluster colors, the performance is poor. During the iteration of K-means, every pixel of the image will be visited. As for a $500 * 500$ image, there are 25,000 pixels, which will cause a slow clustering process. There are various methods to accelerate K-means algorithm, for instance by utilizing the K-D trees to organize the data [KMN*02]. Considering the fact that we are processing colors, we can assign them to different bins. To be clear, we only care about the *a-b* values in the CIELAB color space. The reason why we ignore the L channel value will be explained in Section 3.2. As a result, we will get a $b \times b$ 2D histogram. We use $b = 64$ by default, so that there are only

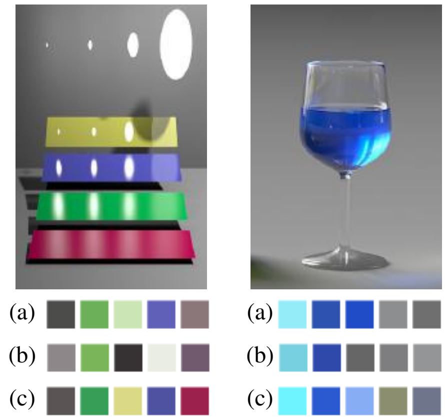

Figure 3: *Results of different automatic palette selection methods. (a) The great result from Chang et al. [CFL*15]. (b) The pleasing result from [HXB*17]. (c) The generated palette of our method is different from (a) and (b). The extracted dominant colors are used to represent the whole image color transformation in the a-b plane, while (a) and (b) never consider this point. It is worth noting that our approach is compared with [CFL*15] and [HXB*17] because they have perfected extracting dominant colors and recoloring images. The left photo is synthesized by Mitsuba [Jak10]. The right photo is from [GZB*13].*

$64 \times 64 = 4096$ pairs in the *a-b* plane. We compute the midpoint in each bin as its representable *a-b* value. In this way, the magnitude of the color number is reduced dramatically. The followed operations in our method are also irrelevant to the size of the source image.

Pelleg and Moore [PM00] noted that the initialization of K-means algorithm is related to it performance. Normally, the initial centers of K-means are selected randomly. However, our intent is to extract enough dominant colors to represent the color relationships in the *a-b* plane. The initial centers should be deterministic as follows. The basic idea is that the colors of different tones are supposed to be the dominant colors even if the number of that color in the image is small. We first calculate the histograms of the hue in the *HSL* color space, while the histogram bin size $b = 18$ by default (we will discuss the selection effectiveness of the dominant color in Section 3.4.) and $0 \leq H \leq 180$. The mean colors in each bin are used as the initial centers of the K-means. However, the calculated cluster centers may not exist in the source image. We wish to extract the colors which are not only in the source image but also can be a representable color in its cluster. Therefore, we recompute the dominant colors by shifting the last-step-computed centers to their neighbors and minimizing the following expression:

$$argmin \sum_{i=1}^{K} \sum_{j=1}^{N} D(C_{dom}, C_j) \tag{1}$$

where K is the cluster number, N is the number of colors in each cluster, C_{dom} is the representable color in its cluster and C_j is the *j-th* color in the same cluster. The function $D(x, y)$ indicates the color difference. In this work, we use Euclidean distance to measure the color difference. The parameter K is determined by the bin size b of the hue histogram. The minimized expression above is solved by iteratively replacing C_{dom} with the shortest-distance point until convergence.

Finally, the dominant colors of the source image are extracted. In Fig. 3, we compare our method with the prior works.

3.2. a-b Color Value Transformation

This section introduces how to map the source colors to the target colors in the CIE-LAB color space. Our method is based on the assumption that the color relationships in the color space should be preserved during recoloring, and the generated results should maintain high fidelity to the source image.

As described in [Alb13], people are expert at differentiating light and dark in the same hue. Using the rods and cones in our eye, we are able to perceive color information, dark and light. The cones are used to perceive color information, while the rods are used to distinguish dark and light. The rods outnumber the cones. The human visual system is sensitive to changes, including high spatial frequency information in luminance [RTB96, War88]. As for visual perception, luminance is more significant than the color [SC03]. Thus, we consider that the color luminance is the key component and keep the luminance unchanged during the color recoloring in order to keep more details of the source image.

We are motivated by the image alignment [Sze06, Sze10], so we consider the color point in the *a-b* plane as the pixel in the image. We define the origin color in the source image as $c = (a, b)$. The target color $c' = (a', b')$ is what we want where f is a function to map c to c' with unknown paramters p. The transformation operates on homogeneous coordinates $\tilde{c} = (a, b, 1)$ and $\tilde{c}' = (a', b', 1)$ can be expressed as $\tilde{c}' \sim H\tilde{c}$ where H is an homogeneous 3×3 matrix. To get matrix H, we need at least four dominant colors. We discuss the influence of the number of dominant colors in Section 3.4.

We assume there is a linear relationship between the amount of motion $\Delta c = c' - c$ and the parameters p as follows:

$$\Delta c = c' - c = J(c)p \tag{2}$$

where $J = \frac{\partial f}{\partial p}$ is the Jacobian transformation f with respect to p. We wish to ensure that each color pair (c, c') will be transformed accurately as much as possible. Thus, we need to minimize the total square residuals as the following expression:

$$E_{TLS} = \sum_{i}^{K} \|J(c_i)p - \Delta c_i\|_2^2 \tag{3}$$

where K is the total number of dominant colors. To get the minimum of E_{TLS}, we get $Ap = b$ by hypothesizing the derivative of E_{TLS} equals to zero where $A = \sum_i J^T(c_i)J(c_i)$ and $b = \sum_i J^T(c_i)\Delta c_i$. To get parameters p, we solve the equation $Ap = b$ by SVD [GVL96]. Furthermore, we employ the power of the RANSAC algorithm [FB87] to improve the transformation accuracy. However, it will jeopardize the objective to maintain as much fidelity to the source image as possible if we naively solve the optimization above; we ignore the relationships between the source palette and the target palette. To solve this problem, we do not allow users to retarget two different colors to the same color. This is acceptable because we want to maintain high fidelity between the source image and our result. Just like image transformation, the color of the source image is transferred successfully through homography estimation. We present detailed inference in the supplemental material.

(a) Source (b) Out-of-gamut (c) Corrected

Figure 4: *The non-linear optimization result of the out-of-gamut image. (a) The source image. (b) The recolored image is out of gamut. (c) The corrected result through non-linear optimization with* $\lambda = 0.3$.

3.3. Non-linear Optimization

This section discusses how to solve the problem that if we directly map the source colors to the target colors through the methods above, some pixels will be out of the gamut because there are some invisible colors which are in the CIE-LAB color space. To solve this problem, we present the non-linear scaling approach to renormalize colors in the *a-b* plane. There are many non-linear functions such as power, logarithm, and exponent. Considering that we need to keep the relationships between colors, we choose the power function. The visible area is different with different levels of luminance. Therefore, we iteratively optimize the colors which are out of gamut to the visible area on each luminance plane. First, we compute the centroid C of the visual area. Second, we iteratively update the distance from each color C' to C non-linearly until color C' is in the visible area as follows:

$$\|CC'_{i+1}\|_2 = \|CC'_i\|_2 - \|CC'_i\|_2^{\lambda} \tag{4}$$

where λ is used to control the step ($0 < \lambda < 1$) and i represents the number of iterations. Initially, C'_i is at where color is out-of-gamut. When the color is near the centroid ($\|CC'_i\|_2 \leq 1$), we will not update the distance because we want to keep the relative distances between the colors and the centroid. During the iteration, we also control the direction $\overrightarrow{CC_i}'$ so that it is unchanged. Fig. 4 shows the non-linear optimization results. The out-of-gamut image in Fig. 4(b) is unsatisfactory, which has many noise points. Fig. 4(c) represents the result that we correct the out-of-gamut pixels. In addition, the result is not sensitive to the parameter λ due to the fact that the colors in the same a-b plane change gradually. The bigger the parameter λ is, the smaller the cost time is.

By default, we let $\lambda = 0.3$ and correct all the pixels locally considering that we care more about the quality of the result rather than the cost in time. We discuss parameter λ in the supplemental material.

3.4. Number Selection of the Dominant Colors

This section discusses how the number of dominant colors affects the recolored result. As for image alignment, the more key points

(a) Source (b) Target (c) b = 40 (d=6.94) (d) b = 30 (d=5.65) (e) b = 20 (d=5.52) (f) b = 10 (d=5.41)

Figure 5: *The recolored result with different parameter b. The d in the parentheses represents the distance to target which is measured as the RMS of CIEDE2000. (a) The source image. (b) The target image. (c) Set parameter b = 40, the shadow of the gem is terrible. (d) Set parameter b = 30, the unexpected colors exist on the gem. (e) and (f) are with b = 20 and b = 10 correspondingly.*

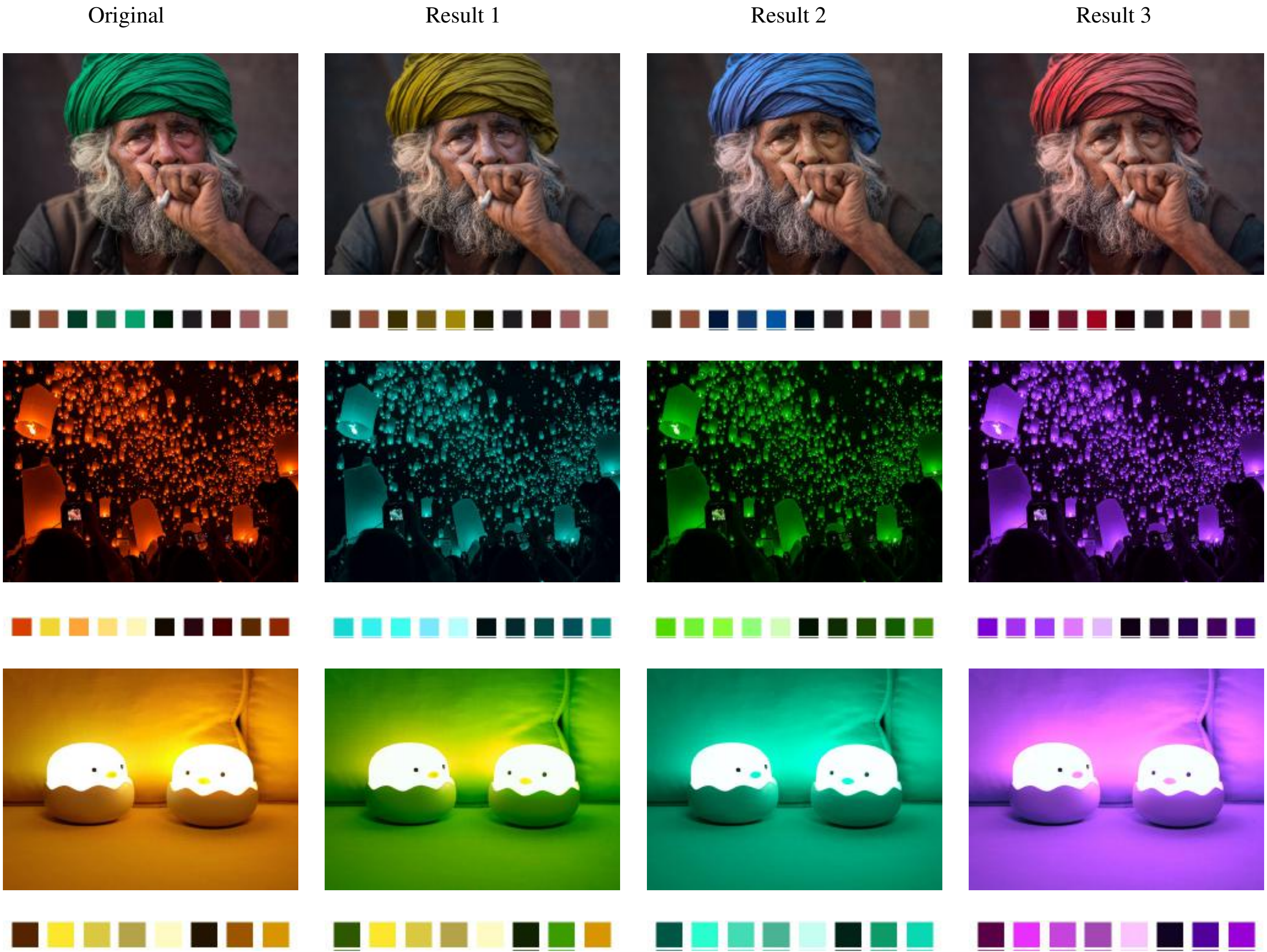

Figure 6: *More examples of recoloring results including both translucent images and normal images. The left column is the original image. The three right columns are recolored results, which include local and global changes with our approach.*

you detect, the more accurate the result is. The key step of all dominant-colors-based methods is to extract the ideal dominant colors. The more dominant colors we extract, the more accurate are the relationships between colors. However, it is not easy for users to edit the color interactively if there are many dominant colors. To solve this problem, we design a prototype interface for users to edit the color easily, which is shown in the supplemental material. In our method, the number of dominant colors is determined by the bin size (b) of the hue. We test the result with different parameters b as shown in Fig. 5. We can observe that the unsatisfactory results appear if b is big, such as Fig. 5(c) and Fig. 5(d). It is worth noting that the CIEDE2000 distance between the result and target in Fig. 5(f) is smaller than Fig. 5(e) although both of them look the same. Considering the balance between the recolored result and the users' interaction, we let parameter $b = 18$.

Through this approach, we can get the state-of-the-art visual results as shown in Fig. 6.

4. Experiments Results

In this section, we evaluate the proposed method through three experiments. The first experiment introduced in Section 4.1 qualitatively compares our methods with the previous work in the literature. Evaluating the palette-based methods remains a challenge. Color evaluation involves personal judgment. No standard translucent image dataset is made in the prior work. First, we either collect free translucent images through the web search engine or purchase translucent images from Shutterstock. Second, we present 3d scene rendering techniques to construct the ground truth to evaluate the color transfer method in Section 4.2. Third, we present a user study to further validate the effectiveness of the proposed method in Section 4.3. Furthermore, we present the results of HDR images in Section 4.4. Finally, we discuss the performance and the limitation of our method in Section 4.5 and 4.6 correspondingly.

Our method is implemented in Python. The results from Chang et al. [CFL*15] are generated through the released online system. The results by Han et al. [HXB*17] are created by the Matlab provided by the authors. The results collected from the baseline methods above are obtained either from the best parameters, which are set by the authors' suggestion or by interactively fine-tuning the parameters to achieve the best result.

4.1. Image Editor Comparisons

In this experiment, we compare our method with the image editor (Photoshop CC) and two other approaches on images which are randomly chosen from the dataset. To guarantee the fairness of the experiment, we provide the same template palettes for all methods. To reduce personal influence, recoloring translucent images by one method is operated by five people. We choose the best result of the five results for each source image.

The representative results are shown in Fig. 7. In each case, our method maintains more fidelity to the original source image than the other approaches. As can be seen qualitatively, it is difficult for humans to separate the shadow area. If the human retargets the color globally, the color of the background will be influenced, as shown in the first row and second column in Fig 7. If the human wishes to retarget the color locally, it will be very difficult to choose the shadow area. Although Chang et al [CFL*15] propose an efficient method that generates state-of-the-art results, it may fail to handle local regions with a small color gap such as the liquor in the fourth row and the amber in the fifth row. They use RBF to interpolate colors. For the shadow area of the image, it is not easy to interpolate colors while maintaining high fidelity to the source image. Then results of [HXB*17] indicate that their method tends to lose some structure information compared to the source image in some limited cases, such as the candle in the second row, while they perform well in other cases, in particular for fabric images. They use decomposition techniques to recolor the image. However, it is difficult to decompose the shadow area in some cases. Unlike them, we recolor the image globally through homography estimation.

Furthermore, we use the gradient information to measure the fidelity of the source image quantitatively. We define the metric as follows:

$$metric = \frac{1}{N}\sum_{i=1}^{N}\left(|G_x(o_i)-G_x(s_i)|^2+|G_y(o_i)-G_y(s_i)|^2\right) \tag{5}$$

where N is the total number of pixels, G_x and G_y describe the gradient information in the horizontal and vertical direction correspondingly, and o_i and s_i represent the i-th pixel in the generated image and source image correspondingly. It can be observed that the smaller the metric value is, the better the result is. In our work, we use Sobel filter operator [SF68] to represent G. We collect 35 test examples in this experiment. The line charts in Fig. 8 describe the comparison results using the metric above. In all cases, the proposed method almost achieves the best results compared with the other methods.

4.2. Ground Truth Comparisons

To further evaluate the color recoloring method, we propose using 3d rendering techniques to construct the original image and ground truth. We generate 30 translucent images with the ground truth through Mitsuba [Jak10] and Appleseed [App11]. The representative examples are shown in Fig. 12. The RMS of CIEDE2000 is used to measure the distance between the recolored result and ground truth. The statistical distance-to-ground-truth result is shown in Fig. 9. The median distance of the proposed method is smaller than that in the works of Chang et al. [CFL*15] and Han et al. [HXB*17].

4.3. User Study

It is always challenging to evaluate a result that includes personal taste. Therefore, we demonstrate that the generated results of the proposed method maintain more fidelity than the other methods through a user study. In this experiment, 40 source images are collected as the test examples, which include the experiment data in the compared literature. The proposed method is compared with the image editor, Chang et al. [CFL*15] and Han et al. [HXB*17].

There are thirty participants who are students or designers in the user study. The fidelity of the source image is evaluated based on three aspects a): structure similarity preservation; b) overall visual effect; c) color harmony. The structure similarity preservation describes the structure similarity between the result and the source image. The overall visual effect describes the overall feeling of the result. The color harmony describes how well the aesthetic of the resulting color is. The participants are asked to rate the results from

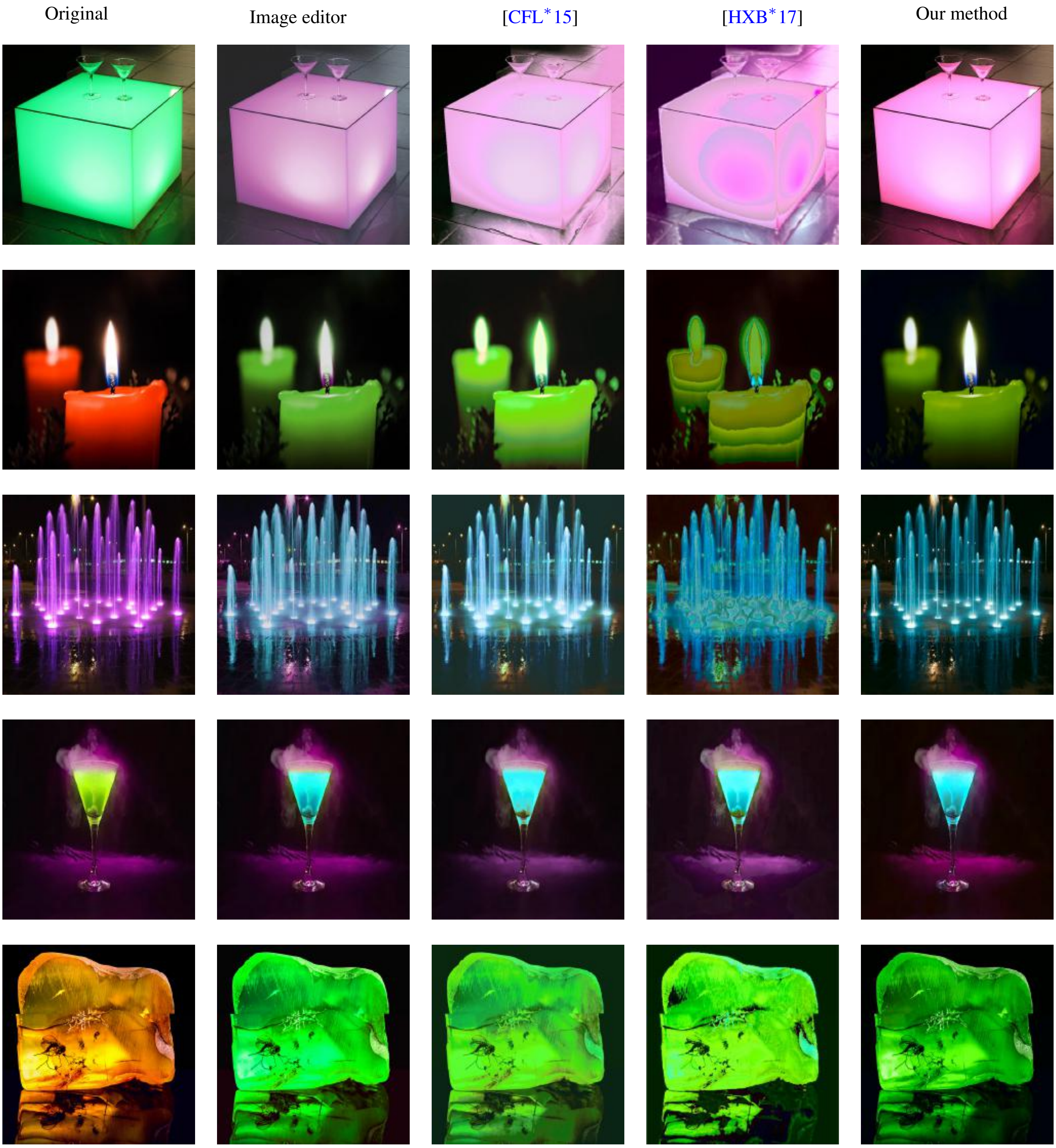

Figure 7: *Comparisons between the proposed method with the image editor (Photoshop CC), Chang et al. [CFL*15], and Han et al. [HXB*17]. The leftmost column shows the original source image. The right columns show image editor (Photoshop CC), Chang et al. [CFL*15], Han et al. [HXB*17], and the proposed method correspondingly. We can observe that the proposed method is similar to the artificial approach. However, this approach requires that the human measure the experience, as the human chooses the shadow area; thus, the color of the background area is influenced, as shown in the second column, if we retarget the color globally.*

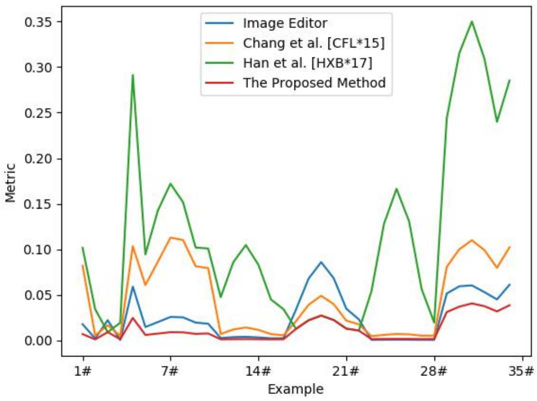

Figure 8: *Comparisons of the gradient preservation result. The smaller the metric value is, the better the method is. We can observe that the proposed method is either better than other methods or almost same as the others.*

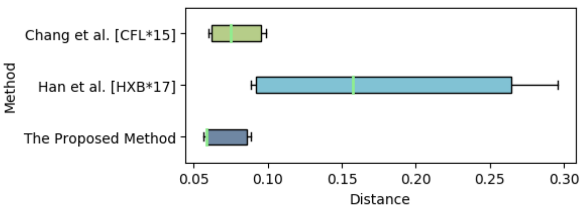

Figure 9: *Comparisons of the distance to the ground truth. The distance is measured as the RMS of CIEDE2000. The smaller the metric value is, the better the method is. The vertical lightgreen line represents the median of the distance. The proposed method is smaller than other methods with statistical significance.*

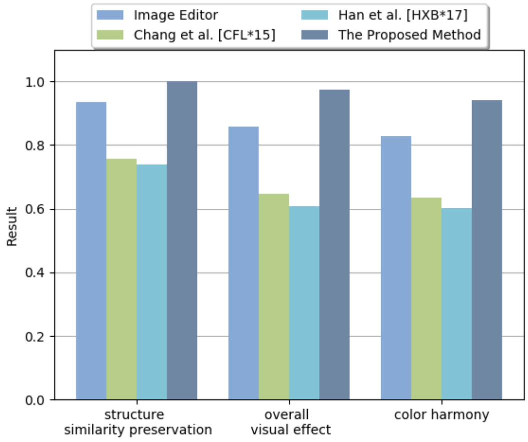

Figure 10: *The overall statistical analysis result. The proposed method is compared with three other approaches. We measure the performance from three aspects: structure similarity preservation, overall visual effect, and color harmony. A higher bar means a better visual performance.*

Figure 11: *The result of recoloring a HDR image. The source image and target image are aligned from left to right correspondingly with their dominant colors. The image is from [DM97].*

1 to 5 for each aspect. We briefly introduce the concept of the evaluated aspects above before the experiment. To avoid bias, we not only arrange the results of different methods randomly but also let users have a rest when they evaluate one image. Via this study, 25 valid surveys are collected.

For statistically analyzing the collected surveys, the ratings of each aspect are first summed up and then normalized by dividing by the number of ratings. The statistical result is shown in Fig. 10. The histograms in Fig. 10 demonstrate that the proposed method achieves better results compared with the three other methods in structure similarity preservation, overall visual effect, and color harmony.

4.4. Recolor HDR images

In this experiment, we try to recolor HDR images as one example shown in Fig. 11. Due to a high dynamic range of luminosity in HDR images, it is still challenging to recolor this kind of image. To be clear, the source HDR images in our experiment are pre-processed by tone mapping by [DD02].

4.5. Performance

There are two primary sources of computational cost in our approach. One is to find the dominant colors of the image. It takes around 2 seconds for a 500×500 image. The other is to map the source colors to the target colors. The homography estimation is fast enough to execute in 1 second for hundreds of dominant colors. If there are colors that are out of the gamut, non-linear optimization takes around 2 seconds to retarget the colors to the visual area in the *a*-*b* plane. In conclusion, after extracting the dominant colors, users can interactively recolor the image. The result of time cost is achieved in Python.

4.6. Limitations

The first step of the proposed method is to extract the dominant colors of the source image. The number of dominant colors is of great significance to the generated result, which is discussed in Section 3.4. It is difficult to determine the optimal number of dominant colors. In the current work, we let the user control the number of dominant colors.

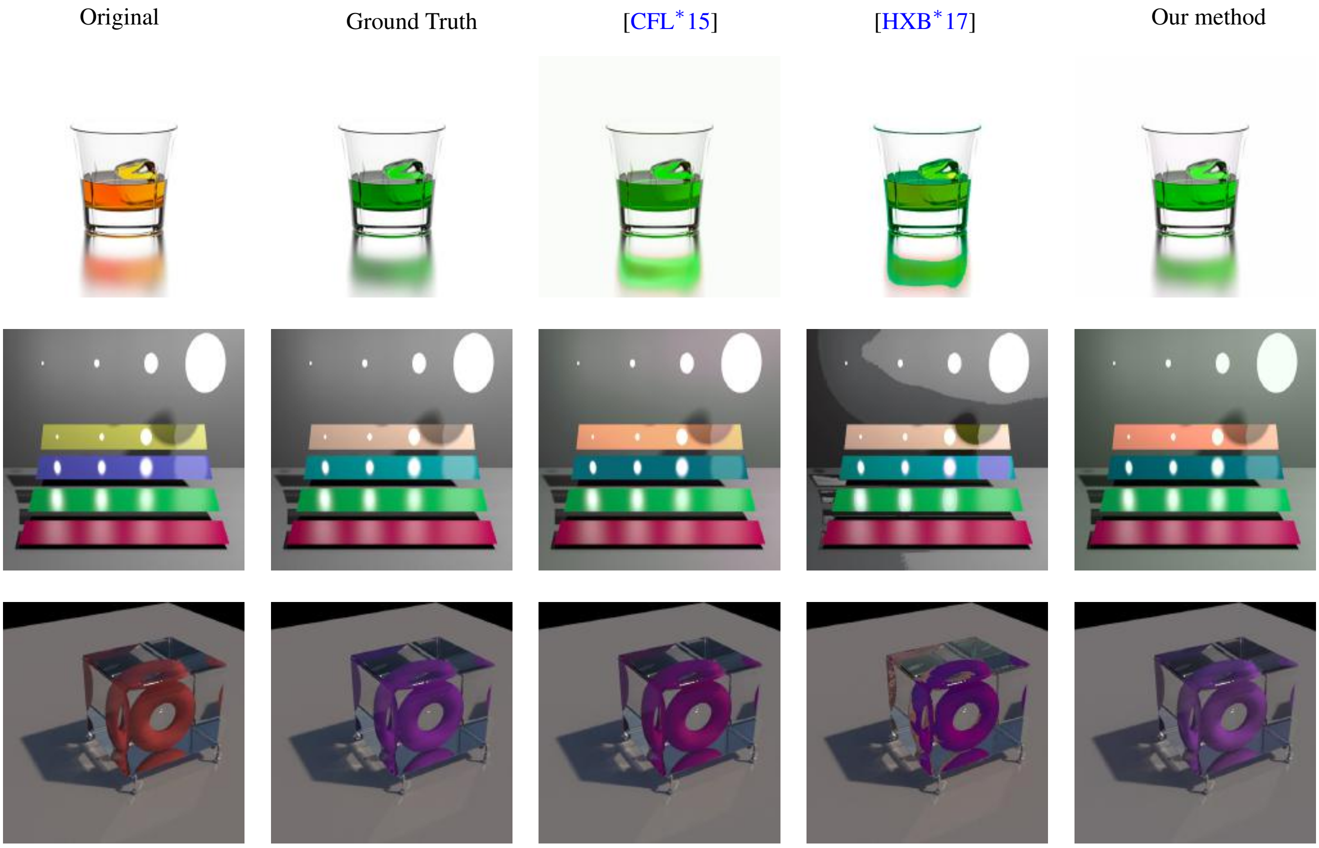

Figure 12: *Some representative comparisons with the ground truth. The original image and the ground truth are generated by the 3d rendering softwares such as Mitsuba [Jak10] and Appleseed [App11].*

In some limited cases, we need to consider the semantic information in the image. For instance, given a source image that consists of a dog and meadow as shown in Fig. 17 of [ZXST17], our method may fail to generate the desired result due to the similar color of the dog and meadow, while the user only wants to recolor the dog. We need to know not only the color relationships but also the pixel location relationships. A powerful segmentation algorithm such as Mark R-CNN [HGDG17] can help a lot in such cases. What's more, how to recolor HDR images on 32-bit channel rather than 8-bit channel is still unexplored, though Section 4.4 presents the results of HDR images pre-processed by tone mapping.

5. Conclusions

In this paper, a new palette-based method to map the source colors to the target colors through homography estimation is proposed. We first present Hmeans method to extract the dominant colors of the source image. Next, the color transformation is performed in the *a-b* plane through the homography estimation. Further, a non-linear optimization approach is proposed to correct the pixels whose colors are out of the gamut. The experiments in recoloring translucent images show state-of-the-art visual results. In particular, the proposed method is able to handle the shadow area of the source image automatically during the color recoloring, which is an improvement compared to previous work.

Image recoloring is a promising and useful research problem. We believe that the proposed approach can help non-experts to recolor translucent images while preserving high fidelity to the source image. However, there are still many challenges in this area. For example, future work will investigate how to recolor the images through transformation in the three-dimensional color space. The semantic analysis and pixel locations of the source image should also be considered during color recoloring. Moreover, automatic target palette recommendations can be added to our recoloring pipeline. Finally, the recoloring approach based on deep learning such as [ZIE16] can also be further modified to recolor the translucent images. However, building a large dataset, which consists of enough translucent images, for deep learning is still a challenge.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grants (No. 61672237, 61532002, 61802128). We wish to thank Dr. Rynson Lau for his discussions in this work. In addition, we thank the anonymous reviewers for their constructive comments that helped us improve the quality of this manuscript.

References

[Alb13] ALBERS J.: *Interaction of color*. Yale University Press, 2013. 5

[AP08] AN X., PELLACINI F.: Appprop: all-pairs appearance-space edit propagation. *ACM Transactions on Graphics (TOG) 27*, 3 (2008), 40. 2

[App11] Appleseed renderer, 2011. https://appleseedhq.net/. 7, 10

[BHW11] BIE X., HUANG H., WANG W.: Real time edit propagation by efficient sampling. *Computer Graphics Forum 30*, 7 (2011), 2041–2048. 2

[BTVG06] BAY H., TUYTELAARS T., VAN GOOL L.: Surf: Speeded up robust features. In *European Conference on Computer Vision* (2006), Springer, pp. 404–417. 3

[CFL*15] CHANG H., FRIED O., LIU Y., DIVERDI S., FINKELSTEIN A.: Palette-based photo recoloring. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 139. 1, 2, 3, 4, 7, 8, 10

[CSUN06] CHANG Y., SAITO S., UCHIKAWA K., NAKAJIMA M.: Example-based color stylization of images. *ACM Transactions on Applied Perception 2*, 3 (2006), 322–345. 2

[CZL*14] CHEN X., ZOU D., LI J., CAO X., ZHAO Q., ZHANG H.: Sparse dictionary learning for edit propagation of high-resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 2854–2861. 3

[DD02] DURAND F., DORSEY J.: Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (TOG) 21*, 3 (2002), 257–266. 9

[DM97] DEBEVEC P. E., MALIK J.: Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 369–378. 9

[FB87] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in Computer Vision*. Elsevier, 1987, pp. 726–740. 3, 5

[GVL96] GOLUB G., VAN LOAN C.: The singular value decomposition and unitary matrices. *Matrix Computations* (1996), 70–71. 5

[GZB*13] GKIOULEKAS I., ZHAO S., BALA K., ZICKLER T., LEVIN A.: Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 162. 4

[HGDG17] HE K., GKIOXARI G., DOLLÁR P., GIRSHICK R.: Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), IEEE, pp. 2980–2988. 10

[HS88] HARRIS C., STEPHENS M.: A combined corner and edge detector. In *Alvey vision conference* (1988), vol. 15, Citeseer, pp. 10–5244. 3

[HSGL11] HACOHEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 70. 2

[HXB*17] HAN Y., XU C., BACIU G., LI M., ISLAM M. R.: Cartoon and texture decomposition-based color transfer for fabric images. *IEEE Transactions on Multimedia 19*, 1 (2017), 80–92. 1, 3, 4, 7, 8, 10

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 4, 7, 10

[KMN*02] KANUNGO T., MOUNT D. M., NETANYAHU N. S., PIATKO C. D., SILVERMAN R., WU A. Y.: An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 7 (2002), 881–892. 4

[LCUR14] LIU Y., COHEN M., UYTTENDAELE M., RUSINKIEWICZ S.: Autostyle: Automatic style transfer from image collections to users' images. *Computer Graphics Forum 33*, 4 (2014), 21–31. 2

[LDLM15] LU S.-P., DAUPHIN G., LAFRUIT G., MUNTEANU A.: Color retargeting: Interactive time-varying color image composition from time-lapse sequences. *Computational Visual Media 1*, 4 (2015), 321–330. 3

[LFUS06] LISCHINSKI D., FARBMAN Z., UYTTENDAELE M., SZELISKI R.: Interactive local adjustment of tonal values. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 646–653. 2

[LH13] LIN S., HANRAHAN P.: Modeling how people extract color themes from images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), ACM, pp. 3101–3110. 4

[LJH10] LI Y., JU T., HU S.-M.: Instant propagation of sparse edits on images and videos. *Computer Graphics Forum 29*, 7 (2010), 2049–2054. 3

[Llo82] LLOYD S.: Least squares quantization in pcm. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137. 4

[LLW04] LEVIN A., LISCHINSKI D., WEISS Y.: Colorization using optimization. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 689–694. 2

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110. 3

[LPSB17] LUAN F., PARIS S., SHECHTMAN E., BALA K.: Deep photo style transfer. *CoRR, abs/1703.07511* (2017). 2

[LRFH13] LIN S., RITCHIE D., FISHER M., HANRAHAN P.: Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 37. 3

[ML09] MUJA M., LOWE D. G.: Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)* (2009), INSTICC Press, pp. 331–340. 3

[NE04] NAZ K., EPPS H.: Relationship between color and emotion: A study of college students. *College Student J 38*, 3 (2004), 396–405. 1

[OAH11] O'DONOVAN P., AGARWALA A., HERTZMANN A.: Color compatibility from large datasets. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 63. 4

[PM00] PELLEG D., MOORE A. W.: X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning* (2000), Morgan Kaufmann Publishers Inc., pp. 727–734. 4

[PR11] POULI T., REINHARD E.: Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics 35*, 1 (2011), 67–80. 2

[QWH06] QU Y., WONG T.-T., HENG P.-A.: Manga colorization. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 1214–1220. 2

[RAGS01] REINHARD E., ADHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications 21*, 5 (2001), 34–41. 2

[RTB96] ROGOWITZ B. E., TREINISH L. A., BRYSON S.: How not to lie with visualization. *Computers in Physics 10*, 3 (1996), 268–273. 5

[SC03] SANCHEZ J., CANTON M. P.: *The PC graphics handbook*. CRC Press, 2003. 5

[SF68] SOBEL I., FELDMAN G.: A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in* (1968), 271–272. 7

[SSCO09] SHAPIRA L., SHAMIR A., COHEN-OR D.: Image appearance exploration by model-based navigation. *Computer Graphics Forum 28*, 2 (2009), 629–638. 3

[Sze06] SZELISKI R.: Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision 2*, 1 (2006), 1–104. 2, 5

[Sze10] SZELISKI R.: *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 5

[TJT05] TAI Y.-W., JIA J.-Y., TANG C.-K.: Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)* (2005). 2

[War88] WARE C.: Color sequences for univariate maps: Theory, experiments and principles. *IEEE Computer Graphics and Applications 8*, 5 (1988), 41–49. 5

[WYW*10] WANG B., YU Y., WONG T.-T., CHEN C., XU Y.-Q.: Data-driven image color theme enhancement. *ACM Transactions on Graphics (TOG) 29*, 6 (2010), 146. 3

[WZL*17] WANG D., ZOU C., LI G., GAO C., SU Z., TAN P.: L0 gradient-preserving color transfer. *Computer Graphics Forum 36*, 7 (2017), 93–103. 2

[XLJ*09] XU K., LI Y., JU T., HU S.-M., LIU T.-Q.: Efficient affinity-based edit propagation using kd tree. *ACM Transactions on Graphics (TOG) 28*, 5 (2009), 118. 2

[XM09] XIAO X., MA L.: Gradient-preserving color transfer. *Computer Graphics Forum 28*, 7 (2009), 1879–1886. 2

[XY*11] XIAO C., YONGWEI N., ET AL.: Efficient edit propagation using hierarchical data structure. *IEEE Transactions on Visualization and Computer Graphics 17*, 8 (2011), 1135–1147. 3

[XYJ13] XU L., YAN Q., JIA J.: A sparse control model for image and video editing. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 197. 3

[YPCL13] YOO J.-D., PARK M.-K., CHO J.-H., LEE K. H.: Local color transfer between images using dominant colors. *Journal of Electronic Imaging 22*, 3 (2013), 033003. 2

[ZIE16] ZHANG R., ISOLA P., EFROS A. A.: Colorful image colorization. In *European Conference on Computer Vision* (2016), Springer, pp. 649–666. 10

[ZXST17] ZHANG Q., XIAO C., SUN H., TANG F.: Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing 26*, 4 (2017), 1952–1964. 2, 3, 10

[ZZI*17] ZHANG R., ZHU J.-Y., ISOLA P., GENG X., LIN A. S., YU T., EFROS A. A.: Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG) 9*, 4 (2017). 3