

Group-Theme Recoloring for Multi-Image Color Consistency

R. M. H. Nguyen¹, B. Price², S. Cohen² and M. S. Brown³

¹National University of Singapore, Singapore

²Adobe Research

³York University, Canada

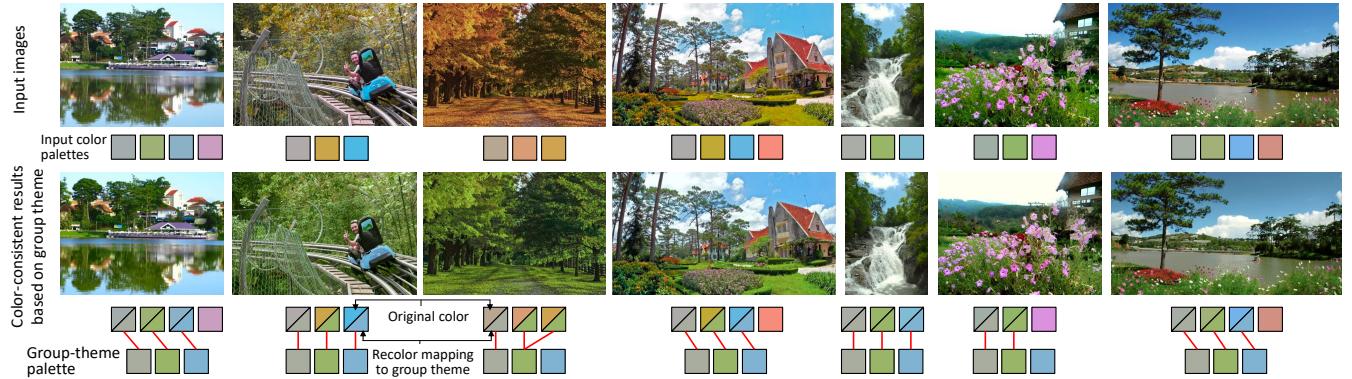


Figure 1: (Top) shows a set of input images that exhibit inconsistent colors. The individual color palettes for each image are also shown. (Bottom) The bottom row shows the images after they have been recolored using colors from a group color theme optimized by our method. The recolored images share a more consistent color appearance.

Abstract

Modifying the colors of an image is a fundamental editing task with a wide range of methods available. Manipulating multiple images to share similar colors is more challenging, with limited tools available. Methods such as color transfer are effective in making an image share similar colors with a target image; however, color transfer is not suitable for modifying multiple images. Approaches for color consistency for photo collections give good results when the photo collection contains similar scene content, but are not applicable for general input images. To address these gaps, we propose an application framework for achieving color consistency for multi-image input. Our framework derives a group color theme from the input images' individual color palettes and uses this group color theme to recolor the image collection. This group-theme recoloring provides an effective way to ensure color consistency among multiple images and naturally lends itself to the inclusion of an additional external color theme. We detail our group-theme recoloring approach and demonstrate its effectiveness on a number of examples.

1. Introduction

It is easy to find images from online resources that can be used for a wide range of purposes, from websites to brochures and slide presentations. The drawback, however, is that it is often hard to find a collection of randomly gathered images that have a consistent appearance in terms of color. This lack of color consistency lowers the aesthetic quality of the final production for which the images are being incorporated. As a result, it is desirable to be able to manipulate the set of images to have better color consistency.

The problem of making images appear more color consistent is not new. There are many methods that aim to manipulate a single input image's colors to appear more consistent with the colors in a reference or target image. This procedure, commonly termed color transfer, is generally limited to a pair of images (source and target). Color editing tools from multiple images have received less attention. The most notable works are methods for editing photo collections. These methods work by finding feature matches among the photo collection. This implies that the images should share simi-

lar scene content (e.g., the same person, the same building, etc). As a result, photo collection editing is not suitable for manipulating a general set of input images. Moreover, these methods are not well designed to incorporate external colors as part of the editing process. The need to consider an additional color theme commonly occurs in applications that already have an existing color scheme, such as brochure or website design.

This lack of color manipulation tools for color consistency of multiple images is the impetus for the framework presented in this paper. Our approach to this problem is inspired by the recent work on color palette manipulation by Chang et al. [CFL*15] and the color theme extraction and exploration abilities of software such as Adobe Color [Ado]. The work in [CFL*15] examined how to effectively represent an image with a small set of colors (i.e., a color palette) and how to recolor the image based on changes to the image’s color palette. In this paper, we use the individual palettes and their recolorization as the underlying mechanism for the color manipulation of multiple images. We describe the necessary components to derive a group color theme and to appropriately assign the group colors back to the individual images’ color palettes. Specifically, we describe a complete framework that allows users to quickly and interactively achieve color consistency across multiple images. In addition, an external color theme can be easily incorporated into the mapping as an additional constraint.

2. Related Work

There are a number of works that address color manipulation that are related to our problem of multiple image color consistency. We discuss them in the following three categories: color transfer, photo collection editing, and palette-based recoloring.

Color Transfer As previously mentioned, color transfer is a well-studied topic in computer graphics with a number of existing methods. These methods aim to modify an input image’s colors such that they are closer to a reference image’s colors. These methods work in either a global or local manner with some additional constraints (e.g., color distribution [RAGS01, TJT05, PKD05], color gradient [PKD07, XM09], tone mapping [HSGL11], or color gamut [NKB14]). Most color transfer methods are automatic, but a few require user assistance to define correspondences between the images [WHCO08, AP10, OHSG12]. There are a few methods that also consider the image’s semantic content in the transfer process [CZG*11, WDK*13, LSL*16]. Given its long history, a full discussion on color transfer methods is outside the scope of this paper and readers are referred to an excellent recent survey by Farid et al. [FPC*16]. While color transfer methods have matured and many produce excellent results, they are not designed for multiple images. When extending pairwise color transfer to multiple images, the obvious solution is to fix one of the images as the reference. Another strategy may be to build some type of composite image (e.g., average of the input images) as the reference. As will be shown in Sec. 4, these strategies do not produce satisfactory results and provide limited control to the user in modifying the color mapping.

Photo Collection Editing There are a number of methods related to editing a collection of images. Similar to color transfer, some of these methods allow an example to be used as reference (e.g.,

[IW05, JMAK10, TISK10]), or allow a personalized editing style to be learned from prior edits (e.g., [KKL10, BPCD11, CKK11]). These aforementioned works target relatively generic editing procedures. Work focused on color consistency includes that of Lafont et al. [LBP*12], who used intrinsic image decomposition applied on photo collections to subsequently allow colored illumination to be consistently applied. Farbman and Lischinski [FL11] perform appearance stabilization in photo sequences from a short video by generating smooth tonal transitions between key frames. Commercial software, such as Adobe Photoshop Lightroom, provides options of automating repetitive tasks by automatically applying a set of recorded editing actions to any number of arbitrary images. According to HaCohen et al. [HSGL13], however, direct copying and pasting of edits without considering image content results in undesirable changes in appearance. To overcome this limitation, HaCohen et al. [HSGL13] showed how to construct a match graph with edges linking photo pairs that share content and then used their prior work on non-rigid dense correspondences [HSGL11] for consistent editing of photo collections. Their method automatically enforces consistent appearance in images that share similar scene content without requiring user assistance. Their results, as well as recent follow-on work proposed by Park et al. [PTSK16], are compelling but are dependent on the image collection sharing similar scene content. In contrast, work proposed by Bonneel et al. [BRPP15] can harmonize a set of images without sharing scene content. Their approach is to transform color distributions of input images to the average color distribution using sliced Wasserstein barycenter. Therefore, their method is dependent on the images sharing similar color distributions.

Palette-Based Image Recoloring There are two notable works [WYW*10, CFL*15] that address color manipulation in an image by using of small number of colors. Wang et al. [WYW*10] presented a data-driven method for enhancing an image based on a specified color theme or mood. In this case, the enhanced image was constrained to the set of colors in the specified color theme (or selected automatically from a database of color themes). Chang et al. [CFL*15] recently presented a method that first extracts a small number of colors, referred to as the image’s color palette, to represent the image colors. Similar palette extraction has been addressed in prior work [SSCO09, OAH11, LH13] and software, such as Adobe Color [Ado]; however, Chang et al. [CFL*15] also introduce a recoloring method that could modify the image’s overall colors based on palette color changes, while minimizing the effects on other colors in the image. Our work is particularly motivated by Chang et al. [CFL*15] and their ability to recolor the image with a small set of colors. Chang et al. [CFL*15] demonstrated that their approach could be used to edit a group of images, however, this was applied to images that were assumed to be color-consistent. They did not explicitly discuss the cases in which input images are inconsistent in terms of color.

In this paper, we present a method that is built on Chang et al.’s [CFL*15] palette-based recoloring as the underlying mechanism for our color manipulation. Our focus, however, is in providing the necessary framework to use this palette-based recoloring technique for multi-image recoloring. Our overall approach is detailed in the following section.

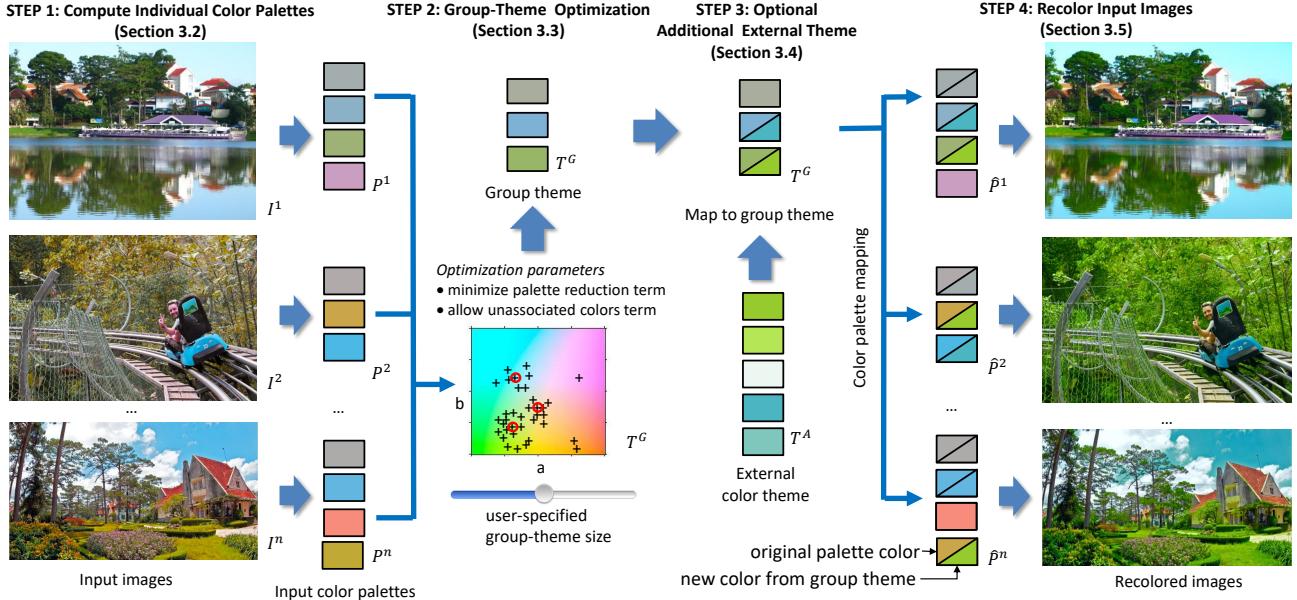


Figure 2: Our group-theme recoloring framework for multiple-images. Step 1: color palettes are extracted for each input image (P^i). Step 2: a group-theme optimization is used to find the group color theme (T^G). The user can set the group-theme size and optimization parameters. Step 3 (optional): an additional external color theme (T^A) can be combined with the group color theme. Step 4: the input images are recolored based on the mappings between T^G , T^A , and P^i . Mappings between color themes and the individual color palettes are denoted by a box with two triangles, where the top triangle shows the original color and the bottom triangle shows the new color. Note while the overall procedure is automatic, the user can interactively change the color mappings as needed (results are updated in real time).

3. Theme-Based Recoloring Framework

3.1. System Overview

Before we begin, we distinguish between the terms color palette and color theme. A color palette is a set of colors used to describe an image. We denote this as P . A color theme, sometimes called a color scheme, is a set of colors that represent the target or constraint for the image collection. We denote this as T .

Fig. 2 provides a diagram that overviews the major components of our framework and their associated sections in the paper. Our framework starts with a set of n input images $\{I^i\}_{i=1}^n$. For each image, I^i , we first extract a color palette P^i that represents the dominant colors in the image. Each pixel in the input image, I^i , is associated with one color in the palette. The details of the palette estimation are given in Sec. 3.2. Next, a group color theme T^G is computed by an optimization function that considers the individual color palettes. The number of colors in the group color theme is specified by the user. The optimization function also serves to assign the individual color palette colors to colors in the group theme. This group-theme optimization is described in detail in Sec. 3.3. The group color theme may be optionally modified by an additional external color theme T^A . This modification is based on the hue values of the external theme and group theme. Details of how to incorporate an additional color theme are given in Sec. 3.4. Finally, to achieve the multi-image color consistency, each image is recolored based on its color palette's mapping to the group theme T^G . This recoloring is described in Sec 3.5.

We note that Fig. 2 does not sufficiently convey the full interactive ability of our software. In particular, our application framework allows the user full control to manipulate the input images, their individual palettes, and the mapping to the group color theme. The user can also dynamically change these mappings, colors in the group palettes, and control aspects of the optimization. For a better appreciation of our software framework's ability, please see the accompanying video.

3.2. Individual Color Palette Extraction

As mentioned in Sec. 2, there are several works that propose methods to extract a color palette from an input image [SSCO09, OAH11, LH13, CFL*15]. Among them, Chang et al. [CFL*15] proposed a fast and robust method based on k -means clustering in the Lab color space. In this paper, we have applied a simple adaptation the Chang et al.'s method [CFL*15] to compute our individual image palettes P^i . In particular, our approach considers only the ab channels of the Lab color space, reducing our problem to 2D clustering. We found that using only ab for both group-theme extraction and recoloring made the application much more intuitive since making the luminance match between multiple images is not a straightforward task and could drastically change the overall brightness of images.

Another issue we also considered is how to choose the number of clusters, k , for each image. The correct choice of k is often ambiguous and depends on the distribution of the data. To address this, we

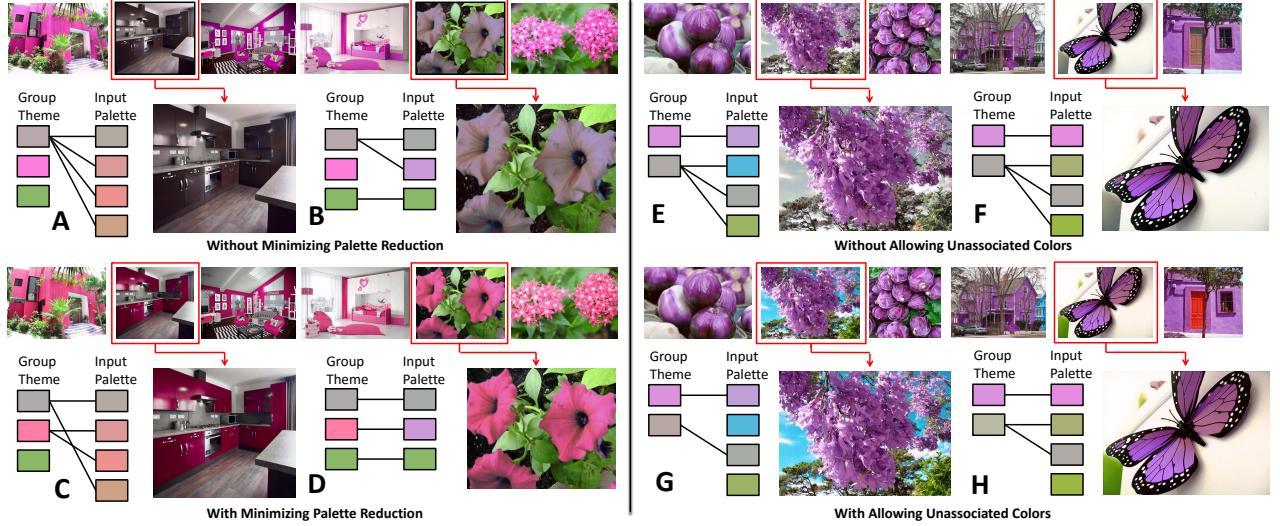


Figure 3: (A & B) show a potential problem with traditional k-means clustering where multiple colors in the input palette map to the same color in the group theme. This has the effect of reducing the color palette of the individual image. (E & F) show another potential problem with traditional k-means clustering where all colors are forced to be assigned to one of the colors in the group theme. (C & D) and (G & H) show how our modified k-means optimization can help overcome the palette reduction and color association problems.

used a technique based on the percentage of *explained variance*. Specifically, we first calculate the *total* distortion v_t by summing the distances between each color point and the overall mean color. Then k-means clustering is performed for different values of k between 2 and 7. For each value of k , the sum of the distance of each point in the cluster to its center (called the *within-group* distortion v_w) is computed. Next, the ratio of the within-group distortion to the total distortion is evaluated. If this ratio is less than a given threshold $\tau = 0.1$, this process is stopped. The current value of k is chosen and the corresponding color palette P is obtained as the centroids of the k clusters.

We apply the algorithm described above on each image I^i to compute the individual color palette P^i of an appropriate number of colors k_i . For each image, we also obtain a histogram w^i , which records the number of pixels associated with each color in the corresponding palette P^i . After this step, we obtain a set of color palettes $\{P^i\}_{i=1}^n$ and a set of corresponding histograms $\{w^i\}_{i=1}^n$.

3.3. Group-Theme Optimization

To achieve color consistency among the multiple images, we seek to compute a group color theme T^G that will be used to modify the individual palette colors as shown in Fig. 2. The size of this color theme, denoted as m , can be controlled by the user and should respond dynamically when changed. The simplest solution is to apply weighted k-means clustering again (where $k = m$) on the set of color palettes $\{P^i\}_{i=1}^n$ to derive the group theme T^G for the entire image set and the assignment for each color in the individual color palette P^i .

For some image collections, this solution is suitable; however, it can have two potentially undesirable drawbacks. The first is that the traditional clustering approach cannot prevent multiple colors in

an individual palette from being assigned to the same color in the group theme T^G . This can be seen in Figs. 3-A–B. In this example, all the palette colors shown in Fig. 3-A are mapped to a single color in the group theme, and two out of three map to the same color in Fig. 3-B. This has the effect of reducing the palette size of the image. The second problem is that simple clustering forces all the colors in the individual palettes to be assigned to the group theme. Such “forced color assignment” can cause a color that is not similar to any of the group-theme colors to still be recolored. This can result in unwanted change in an image’s appearance. This problem occurs more often when the group color theme size is small (e.g., only two colors). Figs. 3-E–F shows an example where palette colors (e.g., the sky in example E and green background in example F) are assigned to a gray color in the group theme.

To overcome these two drawbacks, we propose the following objective function that modifies the traditional weighted k-means clustering method to incorporate two additional terms to: (1) help minimize the problem of palette reduction, and (2) allow unassociated colors. This can be expressed in the following equation:

$$\begin{aligned} \operatorname{argmin}_{T^G, g} \sum_{i=1}^n \left[\sum_{j=1}^{k_i} (1 - \delta(g_j)) w_j^i \|P_j^i - T_{g_j}^G\|^2 \right. \\ \left. + \gamma \sum_{j_1=1}^{k_i} \sum_{j_2 \neq j_1}^{k_i} h(g_j^i, g_{j_2}^i) + \eta \sum_{j=1}^{k_i} w_j^i \delta(g_j) \right], \end{aligned} \quad (1)$$

where the function $h(x, y)$ is defined as follows:

$$h(x, y) = \begin{cases} 1, & \text{if } x = y, x \neq 0, y \neq 0 \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

and the function $\delta(x)$ is defined as follows:

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases}. \quad (3)$$

In Eq. 1, the first term represents the traditional weighted k -means clustering on the set of color palettes, where P_j^i represents the ab value for the j th color in the i th image's color palette. The weights w_j^i represent the number of pixels associated with the j th color in the i th image's color palette. The term $T_{g_j^i}^G$ represents the colors in the group theme, where g_j^i is defined as the assignment of the i th color palette and its j th color to one of the group-theme colors, where $g_j^i \in \{0, 1, \dots, m\}$ and $g_j^i = 0$ means there is no match to the group theme. The second term in Eq. 1 penalizes the cost function when multiple colors in an input palette map to the same color in the group theme. This is controlled by the function h . The third term in Eq. 1 counts the number of individual palette colors that are not associated with the group theme – that is when $g_j^i = 0$. By increasing or decreasing η , we can either enforce color matches or allow for unmatched colors respectively.

Due to the non-convex nature of Eq. 1, expectation-maximization (EM) is used to optimize the function. First, T^G is initialized using the traditional weighted k -means clustering. Then, the algorithm proceeds by alternating between the two following steps, assignment and update, until the assignments no longer change.

Assignment step: Fix T^G , solve for the group assignment g^i for each image I^i . Finding the group assignment for each image can be done independently. In practice, the number of colors in a palette is usually relatively small (e.g. most of graphics designers usually use the palette size of 5). Therefore, a brute-force search can be used to solve this problem in real time.

Update step: Fix g , solve for the new group colors T^G . These new group colors are calculated by the weighted mean of the observations in the new clusters.

After solving Eq. 1, we obtain a group theme T^G with m colors and palette color assignment g_j^i for each image palette's color P_j^i . The user can also control the group-theme result by specifying which of the multiple input images are included or excluded in the group-theme estimation algorithm. An example of this is shown in the accompanying video submitted with this paper.

Parameters and Medoid vs. Mean

We use fixed parameter values for the optimization equation, Eq. 1. Our default setting ($\gamma = 0$, $\eta = 10^{10}$) is used to perform traditional k -means clustering. A large η is necessary to ensure that all palette colors in each image are assigned to the group-theme colors. If the user wants to minimize the color palette reduction, parameters are set to $\gamma = 10^2$, $\eta = 10^{10}$. If the user wants to allow unassigned colors, parameters are set to $\gamma = 0$, $\eta = 10^3$. When both minimizing color palette reduction and allowing unassigned colors are desired, we use $\gamma = 10^2$, $\eta = 25 \times 10^3$.

We also adopt another strategy to improve the final recolored

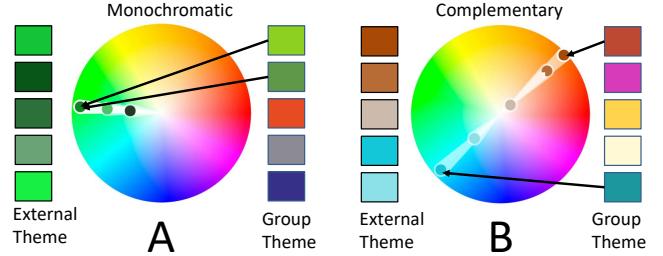


Figure 4: Two examples of additional external color themes, T^A , and their mapping to the group color theme T^G . (A) shows a typical monochromatic color scheme where all colors share the same hue. In this case, only the most saturated color is mapped to colors in T^G . Colors from T^A are mapped to colors in T^G if their hues are within ± 18 degrees. (B) shows a typical complementary color scheme, with multiple colors falling in two distinct hues. As with example (A), only the most saturated color from each hue range in T^A is mapped to T^G .

results. Instead of choosing the mean value of each cluster to be the group-theme color values, we use the medoid color. The medoid is the closest color in the original input palettes to the mean values of the cluster centers. The medoid provides a group theme that avoids introducing new colors that were not in the original input palettes. Therefore, we found the medoid colors give a more natural and vivid result than the mean colors.

3.4. External Color Theme Constraints

For some tasks, such as brochure or website design, a designer often requires an additional external color theme T^A that is used as a color scheme for the background, foreground, and text. In such cases, it is desirable that the accompanying images share some colors in common with this additional color theme. We have experimented with several ways to incorporate the colors of T^A and found that in general, it is not desirable to completely replace the group color theme T^G with all the colors in T^A . Instead, it is effective to incorporate only those colors that are similar to those found in the T^G . We also need to be mindful that many times T^A may use a color scheme that contains colors of a single hue, but at different saturation levels (e.g., a monochromatic color scheme as shown in Fig. 4). We therefore first cluster the additional color theme colors by their hue values. We map the colors in the group theme T^G with those in T^A that share the same hue within ± 18 degrees. If multiple colors in T^A are associated with a hue, we select the color from T^A with the highest saturation level. Two examples of this approach are shown in Fig. 4. There is a chance that none of the colors in the group color theme T^G will be mapped to any of the colors in the additional theme T^A if their hues are all significantly different. In such cases, the user can manually adjust the mapping between T^A and T^G . Similarly, the user has complete control to modify the mapping in the event that they do not like the automatic assignments.

3.5. Image Recoloring

After obtaining the group theme T^G (and any modifications by T^A), we have a mapping between each individual input palette P^i and the group theme T^G . Each input image is recolored by modifying their palettes P^i by the associated color in the T^G . We denote this modified palette as \hat{P}^i . To perform image recoloring, we have modified the method proposed by Chang et al. [CFL*15]. Chang et al. [CFL*15] recolors an input image I based on its original color palette $\{P_j\}_{j=1}^k$ and its corresponding modified palette $\{\hat{P}_j\}_{j=1}^k$. To do so, a set of mapping functions $f_i(\cdot)$ is defined for each pair of the input color P_j and the output color \hat{P}_j . These mapping functions are performed as a translation in the ab channels of the *Lab* color space defined by the change of the output color \hat{P}_j from the original palette color P_j . In their work, radial basis functions (RBFs) are used to compute the weights to blend these function. However, the RBF interpolation is not suitable for handling a large number of values (often in the millions). Although a method to accelerate the interpolation was discussed in their work, we found it necessary to still provide a faster mechanism for recoloring in order to provide interactive rates for our framework. As a result, we use a simplified weight computation by using the inverse distance weighting function which assigns greater influence to palette colors closer to the current color and less influence to those farther away.

Fig. 5 shows two examples that provide comparisons between Chang et al. [CFL*15] and our simplified recoloring version. Fig. 5-A shows the input images and extracted color palettes. Figs. 5-B–C show the results from Chang et al. [CFL*15] and our results with the same changes applied to the color palette. Our results are very similar; however, our simplified version is three times faster than Chang et al.’s approach [CFL*15]. In our approach, we use the simplified recolor approach to achieve interactive manipulation. When all colors have been finalized and the final recoloring is obtained, we can use Chang et al.’s approach [CFL*15] to modify all the input images I^i to obtain their final output \hat{I}^i .

4. Results and Application

4.1. Experimental Setup

As discussed in Sec. 3, our framework takes in a set of images and automatically computes: (1) the individual color palettes; (2) the group-theme colors; and (3) the assignment of the group-theme colors to the individual palettes’ colors. The assigned group-theme colors are used to recolor the images to obtain the results in real time. The group-theme optimization is controlled by the fixed parameters discussed in Sec. 3.3. In addition, our framework is also a fully interactive software that provides real-time visual feedback to the user. In particular, it allows the user full control to examine the input images, the individual palettes, and the mapping to the group color theme. The user can also dynamically change the color mappings, colors in the group palettes, and control aspects of the optimization (e.g., allowing unassigned colors and minimizing palette reduction as described in Sec. 3.3).

Fig. 6 shows several examples produced by our proposed method. The results are arguably more consistent in terms of color appearance compared with the input images. However, in case of input images with palettes that have a number very diverse colors

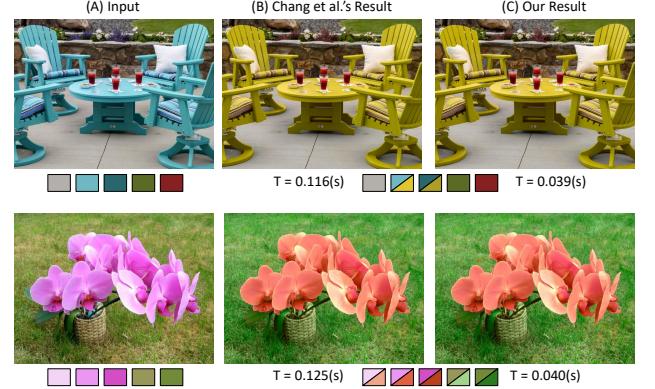


Figure 5: Comparison between Chang et al.’s recoloring method [CFL*15] and ours, showing both the visual results and running time in seconds (s). (A) shows the input image and the extracted color palette. (B) Result of Chang et al.’s method [CFL*15]. (C) Our result. The colors modified in the color palette are shown and are the same for both (B) and (C).

(as shown in Fig. 7-A) taking the mean colors (or mediod colors) may result in a fairly achromatic group color theme. In such cases, the output images may not be as vivid as the input (as shown in Fig. 7-B). In this case, the user can select a subset of input images to be used to compute the group theme. Figs. 7-C–D show two examples of selecting different subsets of images as reference.

4.2. Comparisons with Alternative Approaches

Photo Collection Editing In this experiment, we compare our method with HaCohen et al.’s approach [HSGL13], Park et al.’s approach [PTSK16] and Bonneel et al.’s approach [BRPP15] for photo collection editing. As discussed in Sec. 2, the work proposed by HaCohen et al. [HSGL13] and Park et al. [PTSK16] assumes that corresponding features among the input images can be found. This requires the images to share similar scene content. Therefore, in case of general image groups as shown in Fig. 8 where the images do not share similar content, HaCohen et al.’s method [HSGL13] failed to find any feature correspondences among the images and left the input results completely unchanged. Park et al.’s [PTSK16] method used invalid matches to perform the image manipulation, resulting in more than one image having noticeable color shifts. We note that Bonneel et al.’s [BRPP15] approach produced color artifact since the color distributions between these images were not similar. Our method is able to manipulate the images to appear color-consistent using our group-theme recoloring strategy. In this example, our group-theme optimization was performed with both minimizing color reduction and allowing unassigned colors enabled.

Comparison with Color Transfer We also perform an experiment to show the effectiveness of our approach compared to color transfer methods proposed by Reinhard et al. [RAGS01], Pitie et al. [PKD07], Xiao and Ma [XM09], and Nguyen et al. [NKB14]. As discussed in Sec. 2, color transfer methods are designed to map

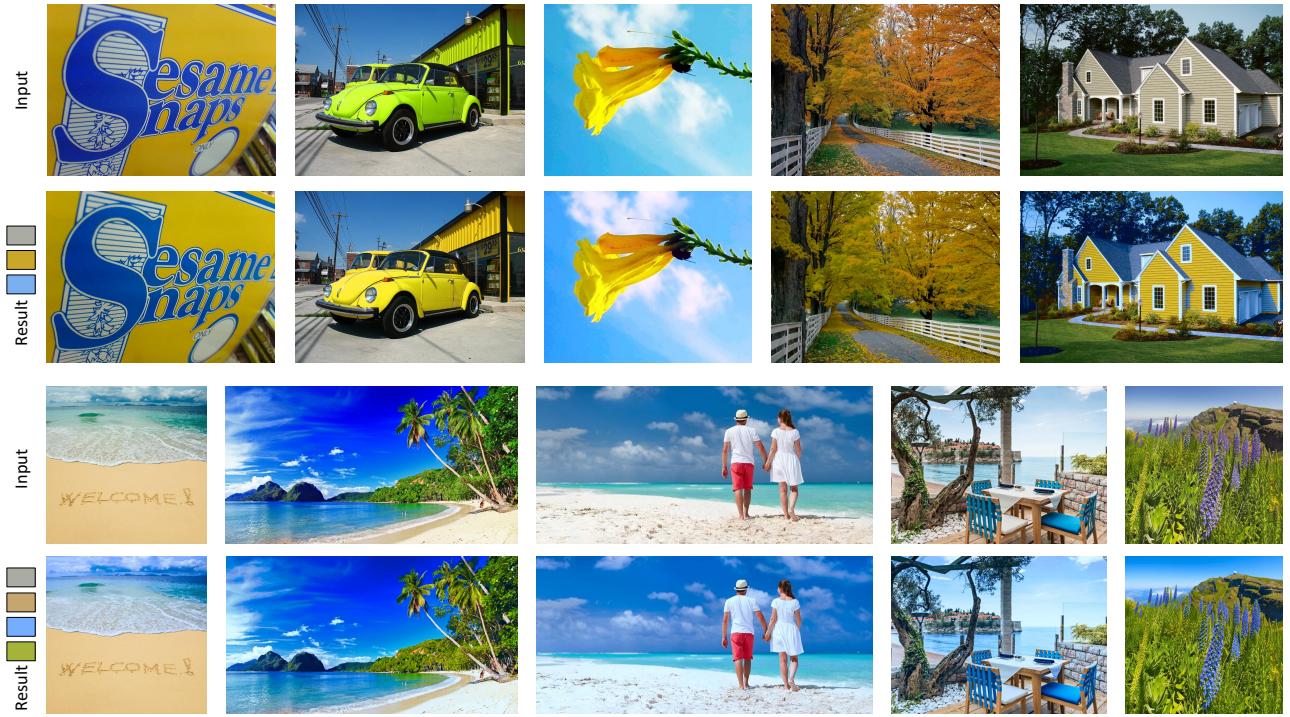


Figure 6: Examples from our results. The first and third rows show the input image with inconsistent colors. The second and fourth rows show our results.

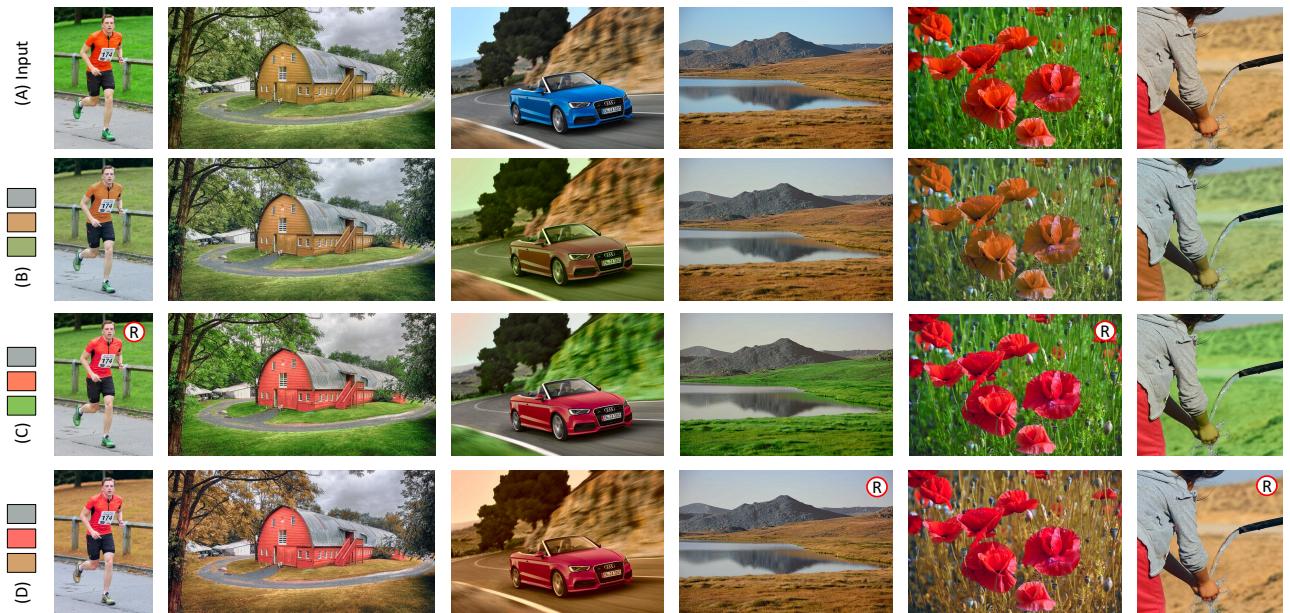


Figure 7: An example of applying our approach with images with more diverse colors. In case of a set containing diverse color scene (as shown in row (A)), using the whole set of images to compute the group color theme may result in greyish colors as they minimize the color differences among the individual palette colors. As a result, the output images may not be as vivid as the input (as shown row (B)). In this case, the user may select a subset of input images as reference. Rows (C) and (D) show two examples of selecting different subsets of images as reference. The images with an (R) overlaid in the upper right corner denote the images selected to compute the group color theme.



Figure 8: Comparison with photo collection consistency methods. The first row shows the input image with inconsistent colors. The second row shows the results obtained by HaCohen et al.’s approach [HSGL13]. Their method failed to find any feature correspondences between any pair of images. As a result, their method produces an output where the images are unchanged. The third row shows the results obtained with Park et al.’s approach [PTSK16] with undesirable color shifts because of mismatched corresponding features. The fourth row shows the results obtained by Bonneel et al.’s method [BRPP15]. This method contains color artifact since the color distributions between these images are not that similar. The last row shows our results.

a source image to a target or reference image. This means that we need to select one of the images from the input image set as the reference image. The remaining images are treated as source images and are mapped to this reference image.

To validate our approach, we performed a user study to determine if our results are preferred over alternative methods. Our user study uses ten sets of images of varying scene content as input. These sets of images were made color-consistent using all comparison approaches (including the above color transfer approaches, Park et al. [PTSK16], and our approach). As part of this study, we also asked a professional designer with over ten years of experience with Photoshop to manipulate the images such that they appear consistent in terms of colors. The expert was given two additional sets together with results from our approach as an example of *before* and *after*. These additional two sets of images were not included in the user study.

The user study had 62 participants, who were asked to select their preferred result between two pairs of results based on two criteria: color consistency and natural-looking. The experiment was carried out in an indoor room with standard fluorescent light. The display monitor used was calibrated to ensure proper color reproduce [Spy]. Each user was presented with 50 pairs of randomly

selected results from all 280 possible pairwise comparisons of the generated results (i.e., $10 \times \binom{8}{2} = 280$). The total number of pairwise comparisons made in the user study was $62 \times 50 = 3100$. This means that each pairwise result for all ten sets of images had at least 10 users give their preference. We used the Thurstone model (Case V) [Thu27] to demonstrate the preference from users and the scale value for each method is shown in Fig. 9 (larger value is better). As can be seen, the most preferred method was the expert’s results; however, our method was a very close second. The other methods are not even close, with the unmodified images more preferred than competing results. One striking difference is the amount of time needed to prepare the results. This is shown in Table 1, where the min, max, and medium number of seconds required by each method are given. Here, each set of images contains from 5 - 7 images with resolution approximate 700K pixels. The expert requires approximately 7–50 minutes to complete a single image set while our approach just takes at most 30 seconds (inclusive of manual adjustments). As a result, our tool provides the ability to produce results on par with an expert but with only a fraction of the time and effort.

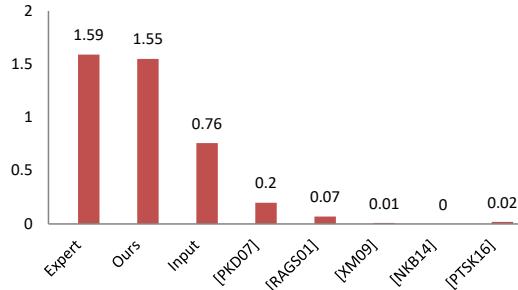


Figure 9: User preference across all methods. The figure shows the scale values of each method using Thurstone model Case V (larger is better). The most preferred method was the expert’s results, however, our method was a very close second.

4.3. Application with Additional Theme

Fig. 10 shows an example that incorporates an additional external color theme into the group-theme recoloring process. The example is a brochure that does not have very good color consistency in its original images (as shown in Fig. 10-(A)). Fig. 10-(B) & (C) show the results where two different additional external color themes are used to modify the original group theme as discussed in Sec. 3.4. These additional color themes are also used to color the brochure itself. As can be seen, our group-theme recoloring makes the images more consistent within the group as well as the brochure theme. Using our recoloring framework, the overall brochure has a more consistent color appearance. In addition, it is easy for the user to adjust the additional theme to obtain a new result quickly.

5. Concluding Remarks

This paper has presented an application framework to achieve color consistency for multiple images. We have shown that recoloring individual input image color palettes based on a group color theme derived from the color palettes provides an effective mechanism to perform this task. This group-theme strategy produces qualitatively more compelling visual results than possible with existing pairwise color transfer methods. In addition, we have shown how external color themes can be incorporated for tasks where additional color constraints are required. To the best of our knowledge, this is the first application tool to allow this type of color manipulation on a group of images.

While our framework makes significant use of the recent image recoloring work by Chang et al. [CFL*15], we use Chang et al.’s recoloring method to implement a novel application that has not been addressed before. As part of this framework, we described several necessary components, including group-theme extraction, an approach for incorporating additional themes, and modification to speed up the image recoloring. As discussed in the paper, one limitation of our approach is not considering images’ semantic content in the group-theme assignment. As a result, some colors can be changed in undesirable ways (e.g. skin tone turns green), although our framework currently allows users to manually adjust unwanted changes. Another limitation of our proposed method is the need to

	Min	Max	Median
[RAGS01]	2	4	2
Ours	6	30	13
[PTSK16]	19	75	28
[PKD07]	38	157	79
[NKG14]	48	201	105
[XM09]	80	404	220
Expert	442	3024	1177

Table 1: Time required by the various methods. The table shows the times (in seconds) for each method in our user study. The expert requires from 7–50 minutes to prepare a color-consistent image set while our approach requires at most 30 seconds.

manually select a set of reference images when the input images are diverse in terms of color. These challenges are interesting topics for future work.

Acknowledgements This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme, a Canadian NSERC Discovery Award, and an Adobe Gift Award.

References

- [Ado] Adobe Color CC. <https://color.adobe.com/>. Accessed: 2016-08-29. 2
- [AP10] AN X., PELLACINI F.: User-controllable color transfer. *Computer Graphics Forum* 29, 2 (2010), 263–271. 2
- [BPCD11] BYCHKOVSKY V., PARIS S., CHAN E., DURAND F.: Learning photographic global tonal adjustment with a database of input/output image pairs. In *The Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), pp. 97–104. 2
- [BRPP15] BONNEEL N., RABIN J., PEYRÉ G., PFISTER H.: Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision* 51, 1 (2015), 22–45. 2, 6, 8
- [CFL*15] CHANG H., FRIED O., LIU Y., DiVERDI S., FINKELSTEIN A.: Palette-based photo recoloring. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 34, 4 (2015), 139. 2, 3, 6, 9
- [CKK11] CAICEDO J. C., KAPOOR A., KANG S. B.: Collaborative personalization of image enhancement. In *The Conference on Computer Vision and Pattern Recognition (CVPR)* (2011). 2
- [CZG*11] CHIA A. Y.-S., ZHUO S., GUPTA R. K., TAI Y.-W., CHO S.-Y., TAN P., LIN S.: Semantic colorization with internet images. *ACM Transactions on Graphics (Proc. SIGGRAPH-Asia)* 30, 6 (2011), 156. 2
- [FL11] FARBMAN Z., LISCHINSKI D.: Tonal stabilization of video. *ACM Transactions on Graphics* 30, 4 (2011), 89. 2
- [FPC*16] FARIDUL H. S., POULI T., CHAMARET C., STAUDER J., REINHARD E., KUZOVKIN D., TREMEAU A.: Colour mapping: A review of recent methods, extensions and applications. *Computer Graphics Forum* 35, 1 (2016), 59–88. 2
- [HSGL11] HACOHEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30, 4 (2011), 70. 2
- [HSGL13] HACOHEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Optimizing color consistency in photo collections. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 32, 4 (2013), 85:1 – 85:9. 2, 6, 8



Figure 10: Application with an additional color theme. (A) shows the input brochure with its original images. (B) and (C) show the results with two different additional external themes are used.

- [IW05] ILIE A., WELCH G.: Ensuring color consistency across multiple cameras. In *The International Conference on Computer Vision (ICCV)* (2005), vol. 2, IEEE, pp. 1268–1275. 2
- [JMAK10] JOSHI N., MATUSIK W., ADELSON E. H., KRIEGLAN D. J.: Personal photo enhancement using example images. *ACM Transactions on Graphics* 29, 2 (2010), 12. 2
- [KKL10] KANG S. B., KAPOOR A., LISCHINSKI D.: Personalization of image enhancement. In *The Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 1799–1806. 2
- [LBP*12] LAFFONT P.-Y., BOUSSEAU A., PARIS S., DURAND F., DRETTAKIS G.: Coherent intrinsic images from photo collections. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012). 2
- [LH13] LIN S., HANRAHAN P.: Modeling how people extract color themes from images. In *SIGCHI* (2013), ACM, pp. 3101–3110. 2, 3
- [LSL*16] LEE J.-Y., SUNKAVALLI K., LIN Z., SHEN X., SO KWEON I.: Automatic content-aware color and tone stylization. In *The Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 2
- [NKB14] NGUYEN R. M. H., KIM S. J., BROWN M. S.: Illuminant aware gamut-based color transfer. *Computer Graphics Forum (Proc. Pacific Graphics)* 33, 7 (2014), 319–328. 2, 6, 9
- [OAH11] O'DONOVAN P., AGARWALA A., HERTZMANN A.: Color compatibility from large datasets. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30, 4 (2011), 63. 2, 3
- [OHSG12] OSKAM T., HORNUNG A., SUMNER R. W., GROSS M.: Fast and stable color balancing for images and augmented reality. In *3DIMPV* (2012), pp. 49–56. 2
- [PKD05] PITIÉ F., KOKARAM A. C., DAHYOT R.: N-dimensional probability density function transfer and its application to color transfer. In *The IEEE International Conference on Computer Vision (ICCV)* (2005). 2
- [PKD07] PITIÉ F., KOKARAM A. C., DAHYOT R.: Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding* 107, 1 (2007), 123–137. 2, 6, 9
- [PTSK16] PARK J., TAI Y.-W., SINHA S. N., KWEON I. S.: Efficient and robust color consistency for community photo collections. In *The Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 2, 6, 8, 9
- [RAGS01] REINHARD E., ADHIKHAMIN M., GOOCHE B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications* 21, 5 (Sep 2001), 34–41. 2, 6, 9
- [Spy] Spyder Display Calibration. <http://spyder.datacolor.com/display-calibration/>. Accessed: 2016-08-29. 8
- [SSCO09] SHAPIRA L., SHAMIR A., COHEN-OR D.: Image appearance exploration by model-based navigation. *Computer Graphics Forum* 28, 2 (2009), 629–638. 2, 3
- [Thu27] THURSTONE L. L.: A law of comparative judgment. *Psychological Review* 34, 4 (1927), 273. 8
- [TISK10] TEHRANI M. P., ISHIKAWA A., SAKAZAWA S., KOIKE A.: Iterative colour correction of multicamera systems using corresponding feature points. *Journal of Visual Communication and Image Representation* 21, 5 (2010), 377–391. 2
- [TJT05] TAI Y.-W., JIA J., TANG C.-K.: Local color transfer via probabilistic segmentation by expectation-maximization. In *The Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 747–754. 2
- [WDK*13] WU F., DONG W., KONG Y., MEI X., PAUL J.-C., ZHANG X.: Content-based colour transfer. *Computer Graphics Forum* 32, 1 (2013), 190–203. 2
- [WHCO08] WEN C.-L., HSIEH C.-H., CHEN B.-Y., OUHYOUNG M.: Example-based multiple local color transfer by strokes. 1765–1772. 2
- [WYW*10] WANG B., YU Y., WONG T.-T., CHEN C., XU Y.-Q.: Data-driven image color theme enhancement. *ACM Transactions on Graphics* 29, 6 (2010), 146. 2
- [XM09] XIAO X., MA L.: Gradient-preserving color transfer. *Computer Graphics Forum* 28, 7 (2009), 1879–1886. 2, 6, 9