DOI: 10.1111/cgf.14964

Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

Integrating High-Level Features for Consistent Palette-based Multi-image Recoloring

D. Xue[1,2,3] , J. Vazquez Corral[2,3] , L. Herranz[2,3] , Y. Zhang[1] and M. S. Brown[4]

[1] Northwestern Polytechnical University, Xi'an, China
[2] Computer Vision Center, Barcelona, Spain
[3] Department of Computer Science, Universitat Autònoma de Barcelona, Barcelona, Spain
[4] York University, Toronto, Canada

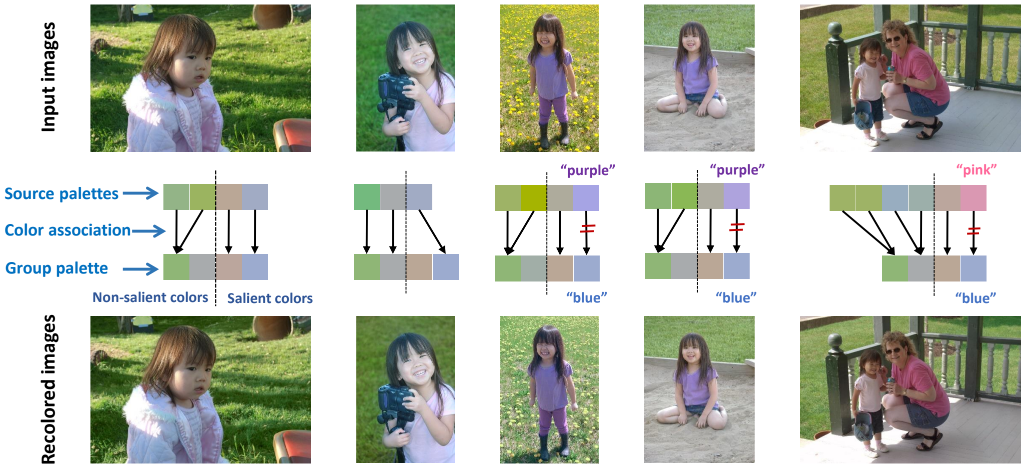


Figure 1: *Our multi-image color consistency framework uses a palette-based recoloring strategy where each input image's palette colors are mapped to a group palette. The estimation of the source palettes, group palette, and their associations are factors in three high-level features: white balance, saliency, and color naming. The recolored collection is visually pleasing and shares visually consistent colors. Top: Input image collection with inconsistent colors. Bottom: Our recolored results. Images are from [BPCD11].*

Abstract

Achieving visually consistent colors across multiple images is important when images are used in photo albums, websites, and brochures. Unfortunately, only a handful of methods address multi-image color consistency compared to one-to-one color transfer techniques. Furthermore, existing methods do not incorporate high-level features that can assist graphic designers in their work. To address these limitations, we introduce a framework that builds upon a previous palette-based color consistency method and incorporates three high-level features: white balance, saliency, and color naming. We show how these features overcome the limitations of the prior multi-consistency workflow and showcase the user-friendly nature of our framework.

CCS Concepts

• ***Computing methodologies*** → ***Image manipulation; Image processing;***

1. Introduction

The need for color uniformity among a collection of images is relevant for applications such as photo collection editing and manipulation of images to have a coherent look and feel for use in websites and brochures. Achieving color consistency among a collection of images is a challenging problem.

Most existing works targeting color consistency focus on transferring colors between a single source image and a target image [TEG18b]. However, these color transfer methods often prove inadequate when dealing with multiple images that require a cohesive color theme. Alternatively, there are methods aimed at editing collections of photos [HSGL13, PTSK16], but they come with specific prerequisites, such as the presence of identical objects (people, buildings, etc.) across the different images for feature matching. Furthermore, these methods are not designed to replace the original set of colors with a completely new color scheme.

In recent years, researchers have leveraged palette-based image recoloring [NPCB17, ZXST17, IY18, DLX*21] to address challenges in multi-image color consistency. These approaches extract a color palette for each input image (source palettes) and generate a combined palette that represents all the images together (group palette). Recoloring is performed by matching individual images' source palette colors to colors in the group palette. Such techniques naturally allow the incorporation of palettes containing colors that were not originally present in the images (*e.g.*, a color palette describing a brochure or website).

Existing palette-based image recoloring methods rely on low-level color statistics and often overlook high-level features related to visual perception. To address this limitation, we propose a comprehensive framework for multi-image recoloring that incorporates state-of-the-art palette-based techniques [NPCB17] and high-level visual features. By integrating these high-level elements, our framework aims to achieve image collections with enhanced color consistency and perceptually natural results. Specifically, we propose to include three modules—white balance correction, saliency-guided palette grouping, and color naming association—to complete the current palette-based multi-image recoloring framework. These individual modules contribute as follows:

White balance correction. Noticeable color differences of photos captured of similar scenes often arise due to different (or incorrect) white balance settings. Strong color casts can adversely affect both individual and group color palette extraction. To ameliorate such color casts, we introduce a white balance correction model to identify and correct the wrong white balance in the image.

Saliency-guided palette grouping. Existing multi-image consistency algorithms often ignore the importance of salient regions when establishing color associations between the source and group palettes. Consequently, while the overall color consistency of the image collection improves, it may lead to inconsistencies in the colors of salient areas due to the influence of non-salient regions. To address this issue, we propose using a saliency module that detects saliency regions to ensure consistent colors across both prominent (salient) and less prominent (non-salient) areas.

Color-naming association. Although we often think of colors sharing similar hues as being visually similar, they can be perceived as distinctly different colors. To avoid unnatural color changes, we introduce a color-naming association procedure, which compares the similarity between the image color palette and the group color palette through the probability of each of the colors belonging to each of the 11 basic color names [vdWSVL09, BVB08].

By incorporating these three high-level features into a multi-image consistency framework, we showcase significant improvements over relying solely on low-level color statistics. To validate the effectiveness of our framework, we conduct a user study that demonstrates a strong preference for images recolored using our approach.

2. Related work

Related works are presented for the following: photo-collection editing, palette-based recoloring, color naming, and saliency-aware image editing.

2.1. Photo-collection editing

Popular image editing software like Adobe Photoshop and Lightroom offer batch processing functionality to apply editing operations to an entire image collection. However, this approach often falls short in adapting to the collection's varying scene content and lighting conditions.

Many prior color consistency approaches leverage shared content among the image coloration, such as recurring architectural structures or people. These methods employ techniques to adjust color transformation curves [HSGL13, PTSK16, XYX*17] or optimize white balance and gamma correction parameters [PTSK16] to enhance color consistency across the collection. Such methods, however, can be ineffective when the input collection lacks common content. To address this challenge, Nguyen *et al.* [NPCB17] introduced a palette-based framework for generic multi-image recoloring to adapt to different image contents within a collection. Their method focuses on palette manipulation, allowing users to intuitively adjust image colors. By operating in the *Lab* color space, it effectively avoids the issue of over-saturation. In a similar vein, addressing the challenge of scene changes in videos, Du *et al.* [DLX*21] proposed a 4D skew polytope with a limited number of vertices. This polytope serves as an approximate enclosure for video pixels across color and time dimensions, implicitly defining time-varying palettes.

In this paper, we build upon the framework proposed by Nguyen *et al.* [NPCB17]. Our key contribution is to incorporate high-level features into Nguyen *et al.*'s basic framework. As described in Sec. 1, these high-level features focus on white-balance correction, salient region considerations in individual palette extraction, and color naming as a way to make associations between individual and group palettes.

2.2. Palette-based image recoloring

The palette-based image recoloring approach was introduced by Shapira *et al.* [SSCO09]. This approach simplifies image manipulation by summarizing an image with a small set of colors (a color palette). This technique allows for easy adjustments to the image by modifying the individual palette colors (*i.e.*, changing a palette color to a new color). This straightforward, user-friendly approach does not rely on extensive professional knowledge or reference images. The technique's success depends on extracting a good representative palette and adjusting the palette colors correctly. Palette extraction methods are typically categorized into two types: clustering-based and geometry-based. Clustering-based methods [CFL*15, ZXST17] determine palette by the frequency of color occurrence. Geometry-based approaches [TLG17, TEG18a, TEG18b, WLX19] construct convex hulls in various color spaces, with the vertices serving as the palette colors.

Clustering algorithms for palette extraction can be adversely impacted by strong color cast due to scene illumination (*i.e.* white balance) and locally distinctive colors with low occurrence. To address the illumination problem, Iwasa *et al.* [IY18] and Liu *et al.* [LZXD21] perform intrinsic color decomposition and limit recoloring to the reflectance image. However, inaccurate decomposition methods can adversely affect the final recoloring result, and recoloring only the reflectance image may not be intuitive as the color of the composed image changes when the illumination image is combined. Some k-means-based methods utilize color histograms for clustering, which may overlook small but significant colors, resulting in a non-representative palette. Kang *et al.* [KH18] enhance palette extraction by computing patch uniformity for local patches.

In contrast to the methods described earlier, our framework has a distinct objective of achieving color consistency across a complete set of images by merging the palettes of each individual image. To address the challenge of varying color casts due to illumination among images, our framework incorporates a white balance correction module to remove strong color casts from images when needed.

2.3. Color naming

Color names are the words used to describe and differentiate colors. Color naming systems can vary across cultures and languages. Seminal work by Berlin and Kay [BK91] found that most societies and cultures share a set of 11 linguistic distinct color names: *red, orange, brown, yellow, blue, pink, purple, green, black, gray, white*. Color naming is crucial in design, industries, and vision research. Recent color naming models involve probabilistic graphical models [HS12, LYC*15], deep neural networks [YCvdW18], and statistical approaches [BVB08] to map physical color stimuli to corresponding color names. The goal is to improve the accuracy and consistency of color naming predictions. The representations of the color name are also constantly being expanded, from the primary 11 colors to more detailed color classifications [YZvdW*18]. Other studies [MHJP18] are exploring the cultural and linguistic factors influencing color naming systems across different languages and cultures.

Color names can be considered features that are connected to human perception. By ensuring the consistency of color names between the source color and target color, we can mitigate unnatural color transitions to a certain degree.

2.4. Saliency-aware image editing

Salient objects of an image are those on which our attention focuses first. In particular, the salient part of the image stands out from its surround because of a difference in one or more physical factors, discontinuities, or lack of correlation [KML18]. Saliency is widely used in image editing based on the natural perception difference of human vision to different regions of the image. The key details and saliency structures can be preserved by distinguishing the optimization target of salient and non-salient areas, improving the perceptual visual quality. The different methods differ in how they focus on highlighting the salient region desired by users: color transferring [VCB17, MSZ19, MBK*23], cropping [TNZ*20], or object removal [JXWS21].

In this paper, we propose using saliency as a reference to differentiate prominent colors in salient areas from those in other regions. This strategy categorizes palette colors as belonging to salient or non-salient regions. When applying the group palette to recolor individual images, the two categories can be treated separately, minimizing the influence of improper color combinations on the salient objects in the image.

3. Our multi-image recoloring framework

Consider a collection of images $\{I_s^i\}_{i=1}^n$, where n represents the number of images. The objective of multi-image recoloring is to obtain a set of recolored images $\{I_t^i\}_{i=1}^n$ that retain the same content but exhibit improved perceptual color consistency.

Palette-based image collection recoloring typically encompasses three primary steps: (1) extraction of source palette for each image, (2) generation and matching of a group palette to each image, and (3) recoloring the images based on the palette adjustments.

In the initial step, we aim to derive a source palette for each image, denoted as $P_s^i = \{c_s^1, c_s^2, ..., c_s^{k^i}\}$. These palettes comprise the primary colors specific to each image. The source palette is extracted by k-means clustering. The number of cluster centers k_i is determined by the percentage of explained variance, calculating the ratio between the total distortion and within-group distortion for different k_i values. During the process, the k-means clustering is performed varying k_i from 2 to 7. The optimal k_i value is chosen when the ratio is lower than 0.1, and the cluster centers are the source palette colors. Next, we proceed to the second step, where a unified group palette for the entire image collection, denoted as $Pg = \{c_g^1, c_g^2, ..., c_g^{k^g}\}$, is generated. The group palette is determined using a weighted k-means clustering method that incorporating two additional terms [NPCB17]. The first term aims at avoiding palette reduction, where multiple colors in the source palette are assigned to the same color in the group palette. The second term aims at accommodating unassociated colors, that is, colors in the source palette that are not assigned to the group palette. An optimization is performed until a group palette is obtained and source color associations to the group palette no longer change (see [NPCB17] for more details).

Finally, this matching solution is then utilized to produce the recolored images. Color mapping is performed in the *Lab* color space's *ab* channels, determined by the matching between each source palette color c_s and its corresponding group color c_g. The weights are determined by an inverse distance weighting function, assigning larger weights to closer palette colors.

The process described above relies solely on the statistical color information present in the images, enabling its applicability to image collections encompassing diverse content. However, it is important to acknowledge that color perception extends beyond pixel-level attributes [Fai13] and that higher-level cues significantly shape our perception of color [OHG12].

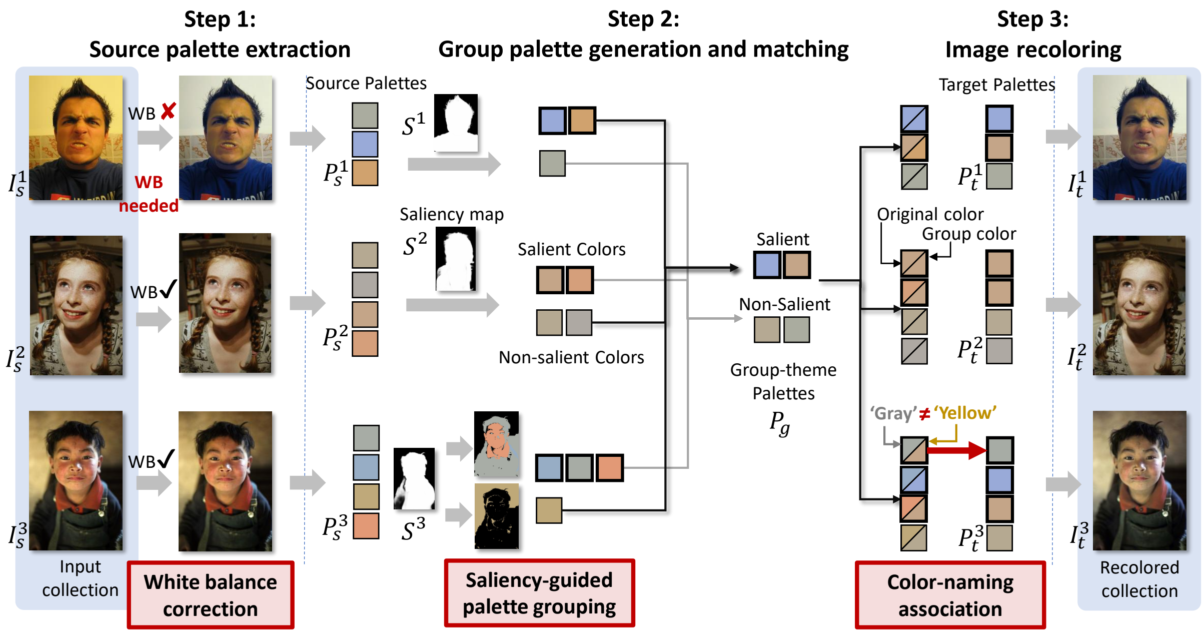


Figure 2: *Our image collection recoloring framework. Given the input collection of images, I_s^i. Step 1: the source palette P_s^i of each image I_{wb}^i is extracted by k-means clustering. White-balance correction (either automatic or manual) is applied before the clustering procedure. Step 2: Colors in each source palette P_s^i are categorized based on the saliency map S^i into two groups: a sub-palette for the salient regions, and a sub-palette for the non-salient regions. The group palette P_g is computed based on the salient and non-salient source palettes of all the input images, respectively. The colors in each source palette match the color in the group palette. Color associations between the source and the group palette with inconsistent color names are removed. Step 3: The images are recolored based on the mapping between source palette P_s^i and group palette P_g.*

We present a comprehensive framework to address the limitations of existing multiple image recoloring approaches (illustrated in Fig. 2). As previously discussed, our framework incorporates three modules based on high-level features into the recoloring processing: white-balance correction, saliency-guided palette grouping, and color-naming association.

3.1. White-balance correction module

White balance is a critical process applied by digital cameras. White balance aims at mimicking the color constancy ability of the human visual system. This ability allows us to perceive the color of an object the same, even when viewed under different illuminations. For example, we can perceive a sheet of paper as "white" under yellowish tungsten or bluish outdoor light. Significant research efforts have been dedicated to developing white-balancing methods within camera pipelines [Buc80, GGvdW11, VCVBT12, HWL17, BT17]. However, relatively little attention has been directed toward addressing the issue of enhancing images with incorrect white balance.

The presence of incorrect white balance can significantly impact the overall color distribution of an image. This, in turn, poses challenges when using palette-based image recoloring techniques, as the extracted source palette and group palette may be biased towards the illuminant color (refer to Fig. 2). Our framework incorporates a pre-processing step to address images with strong color casts to ensure a collection with natural-looking colors. Specifically, we employ the method proposed by Afifi *et al.* [APCB19]. In our framework, we allow users to manually select which images undergo white balance correction and which do not. For the automatic processing of images, we evaluate the color difference for all the images in the input collection to determine whether white balance correction should be applied. In particular, we compute the following metric:

$$\Delta E = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\left(L_s^i - L_{wb}^i\right)^2 + \left(a_s^i - a_{wb}^i\right)^2 + \left(b_s^i - b_{wb}^i\right)^2}, \quad (1)$$

between the original input image $I_s(L_s, a_s, b_s)$ and the image after white-balance correction $I_{wb}(L_{wb}, a_{wb}, b_{wb})$, where n is the number of pixels of the image. If the average color difference ΔE of all the pixels in the image is larger than the set threshold d_{wb}, we consider the white balance of the original image to be inaccurate. In this case, we use the white balance corrected image as input for the subsequent processing. Note that the white balance correction is only applied to images deemed improperly white-balanced since well-white-balanced images do not impact the overall result, and therefore, leaving them uncorrected has little influence on the final output.

3.2. Saliency-guided palette grouping module

Salient regions play an important role in the initial stages of our visual system, as they are prioritized for further processing in the visual cortex, shaping our overall understanding of an image. Dur-

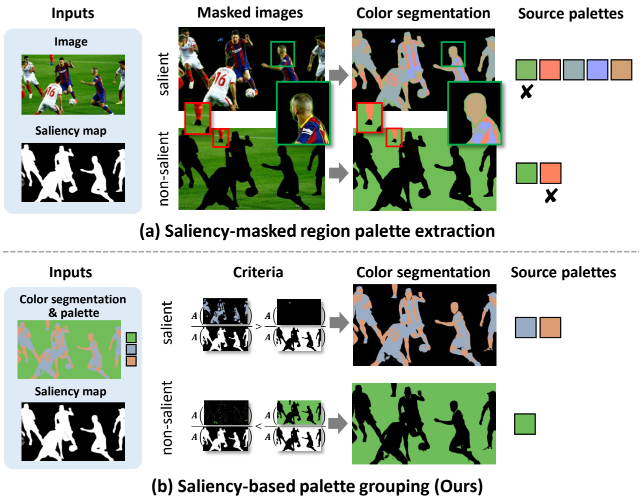


Figure 3: *Saliency-based source palette grouping. The method in (a) uses the saliency map as a reference to extract the palette from the salient and non-salient regions, which may produce similar colors. (b) Our approach combines the saliency map and color segmentation to obtain palettes of salient colors, reducing the impact of inaccurate edges and missing areas in the saliency map.*

ing the process of group recoloring, it is important to try to preserve salient regions, even though their colors contribute to only a small portion of the images. Failure to do so can lead to a deviation in the perceived collection of images from the intended representation.

We introduce a module that generates a content-aware group palette, utilizing the saliency map as a reference. Specifically, we categorize the source palette into two distinct sub-palettes: the salient palette $P_{s,salient}$ and the non-salient palette $P_{s,non-salient}$. One approach to obtain these sub-palettes is by extracting palettes separately from the image's salient and non-salient regions, respectively. This can be achieved using the masked image regions, as illustrated in Fig. 3 (a). However, this method may result in overlapping foreground and background palettes due to deficiencies in the saliency map. These deficiencies can include inaccurate edge segmentation and the presence of spurious regions.

To avoid such artifacts, we first extract the source palette by k-mean clustering following [NPCB17] and assign each pixel of the image with a color from the source palette to get a color segmentation map L. The value of each pixel in L represents the color of its corresponding cluster center. As shown in Fig.3 (b), the sub-palettes are then decided by comparing the proportions of the number of pixels with the same color label in the salient and non-salient regions using the following expression:

$$\begin{cases} c_s^k \in P_{s,salient}, & \frac{\mathscr{A}(L=c_s^k \cap S>\gamma)}{\mathscr{A}(S>\gamma)} > \frac{\mathscr{A}(L=c_s^k \cap S\leqslant\gamma)}{\mathscr{A}(S\leqslant\gamma)} \\ c_s^k \in P_{s,non-salient}, & \text{elsewhere} \end{cases}, \qquad (2)$$

where $\mathscr{A}$ indicates the area of the region. L is the color segmentation map, where $L = c_s^i$ indicates the region with color c_s^i. The saliency map S encodes a per-pixel probability of that pixel belonging to the salient region. To distinguish between salient and non-salient pixels, we use a threshold γ. Specifically, when $S > \gamma$, it indicates the presence of salient regions. This approach effectively assesses the importance of colors in different areas and facilitates the separation of salient and non-salient colors. In our framework, we use a CNN-based saliency object detection method [WWW*20] to obtain the saliency map—alternative approaches could also be used.

After organizing the original palette into salient and non-salient colors, the group palette is generated for each category. This avoids inconsistencies in the colors of salient and non-salient areas due to the influence of another region. The matching process between each image's source palette and the group palette is performed considering both salient and non-salient colors. This matching can be done in two different ways.

- **Both salient and non-salient:** To color match the salient and non-salient parts of the palette separately. In this case, our saliency module performs consistency in the salient and non-salient colors separately.
- **Non-salient only:** To only color match the non-salient while allowing the salient part to be kept as it was originally. In this way, salient regions are left unmodified, while the images' background is consistent.

Note that we use the first approach unless mentioned otherwise. The second method is suitable for some special scenarios where users want to maintain the diversity of the salient area.

3.3. Color-naming association module

In specific scenarios, there may be noticeable differences between the colors selected in the source palette and the group palette, resulting in unnatural recoloring. For example, in Fig.8, the colors of the feather (column 1) and the clouds (column 3) underwent significant changes. These unnatural color transformations are easily noticeable to observers due to their large shifts in hue [LPU*13].

To address these issues, we propose a module in the group recoloring process that considers color naming. Specifically, our approach constrains the recoloring to only those colors that share the same color name based on the 11 basic color terms (red, orange, brown, pink, purple, yellow, green, blue, black, grey, white) [BK91]. This means that an orange color from the source palette, for instance, will not be transformed into a yellow color in the group palette. By incorporating color names as relevant perceptual features to assess the color differences between the matched colors in the source and group palettes, our aim is to maintain perceptual consistency between the input image and the recolored image. Compared to directly applying a threshold in perceptual color spaces, such as *Lab*, our approach greatly minimizes unnatural hue shifts with the incorporation of color-naming associations.

We employ a color naming identification model to determine the association between the source palette color c_s^i and the corresponding matched group palette color c_g^j. This model generates a probability vector indicating the likelihood of each RGB color belonging to specific color names (specifically, we utilize the model described in [vdWSVL09]). Subsequently, we calculate the Euclidean distance between the probability distributions of the source color $p_{c_s^i}$ and the group palette color $p_{c_g^j}$. If this distance exceeds a certain

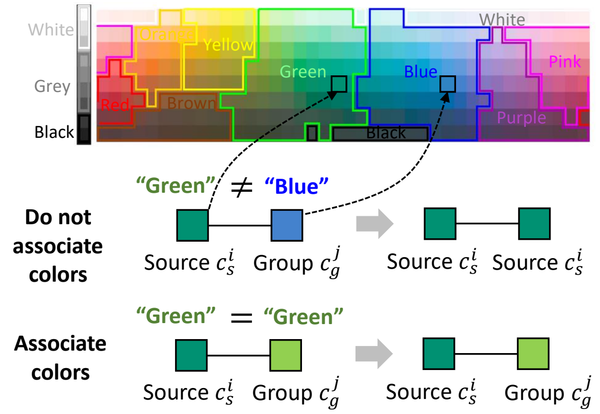


Figure 4: *Color-naming association. The matched source color and group color are not associated if they have different color names.*

threshold, denoted as d_{name}, we consider the color match inappropriate; therefore, the source color remains unchanged. By adjusting the value of d_{name}, we can control the extent of color modification in the recolored image as follows:

$$\begin{cases} \text{associate match } (c_s^i, c_g^j), & \left\| p_{c_s^i} - p_{c_g^j} \right\|_2 \leqslant d_{name} \\ \text{do not associate match } (c_s^i, c_g^i), & otherwise \end{cases}. \tag{3}$$

An example of this procedure is shown in Fig. 4.

4. Results and applications

4.1. Experimental setting

As previously described, our framework adds three modules to the basic framework proposed by Nguyen *et al.* [NPCB17]. The modules can be used independently or together to solve the hard cases in multi-image recoloring. The framework can be run in automatic mode, or interactively to obtain different recolored collections for different needs.

All results presented in this section were obtained using our automatic processing. To ensure a fair comparison, we limited the number of group colors and kept the parameters for our model unchanged. Specifically, we set the values for d_{wb}, γ, and d_{name} to 20, 0.9, and 0.8 respectively. For the automatic process, the number of group palette colors is the average of the number of source palette colors. Thresholds used in the automatic process were chosen experimentally through parameter search (visually guided grid search). While adaptive variables could be a better way, implementing them is challenging since some inputs will inevitably require a user in the loop. Note that our interactive framework also allows the user to manually select images for white-balance correction, adjust the thresholds for each module, and apply different group palette numbers. The images used for these experiments were sourced from the MIT-Adobe FiveK Dataset [BPCD11] and Flickr [Fli04]. Our multi-image recoloring framework is implemented with Python, and the user study computations are implemented using MATLAB's Psychotoolbox.

We compare against the two versions presented in the Nguyen *et al.* algorithm: the basic framework and the "unassigned" version. The latter one breaks connections among colors by penalizing large distances between the source and group colors. In both cases, we use the same parameters as proposed in [NPCB17].

4.2. Qualitative results

White-balance correction module. Color cast due to incorrect white balance is particularly noticeable when images contain people, as our perception is highly sensitive to the appearance of faces. An example showcasing the effectiveness of our white-balance module is illustrated in Fig. 5. In the first row, the original images are displayed in (a)-(e). In the second and third rows (versions of Nguyen's method), the dominant yellow color in (e) influences the palette extraction, resulting in the exclusion of the blue colors present in (a) and (e) from the palette selection. As a consequence, non-realistic outcomes are produced for these two images. However, this problem is resolved by incorporating our white-balance correction module. In particular, a significant improvement can be observed in the last column of the first row. This correction enables the group palette to represent the blue colors present in the images better, leading to a more consistent set of recolored images that exhibit natural and vibrant tones across different skin tones—as shown in the last row.

Saliency-guided module. Our method can distinguish salient and non-salient areas even when the saliency map is inaccurate. This is shown in Fig. 6, where the saliency map—shown in the second row—does not completely distinguish the salient objects (in this case, the buildings), but our salient colors—shown in the third row—do correctly distinguish them. The results obtained by our method—last row—improve due to the correct categorization of the salient and non-salient regions at the palette level. In this way, the salient color obtained with our approach covers the common architecture, while the non-salient one covers the sky region.

Fig. 7 provides an illustrative example of the two types of matching options that can be performed using saliency information. Looking at the row that presents the result for the method that matches both salient and non-salient regions, we can observe that all the flowers (salient regions) are modified to a more consistent red color, while the leaves (non-salient regions) are recolored into a more consistent green color. The results show that the separation of salient and non-salient palettes avoids inconsistent color caused by unwanted associations between salient and non-salient colors. For the standard version of Nguyen's approach without separation, the salient region in Fig. 7 (a), (c), and (e) will produce inconsistent recolored results due to the association with the non-salient color palette green. On the other hand, in Fig. 7-last row where only the non-salient region undergoes consistency adjustments, our method results in a more uniform green color for the leaves, while the flowers retain their original colors. For this non-salient-only setting, the unassigned version of Nguyen's approach achieves a result similar to our last case. Nevertheless, it is worth mentioning that Nguyen's approach relies on the color differences in the *ab* channels of *Lab* space, making it difficult to unassign unnatural color changes, as will be further explained in the subsequent module and user experiments.

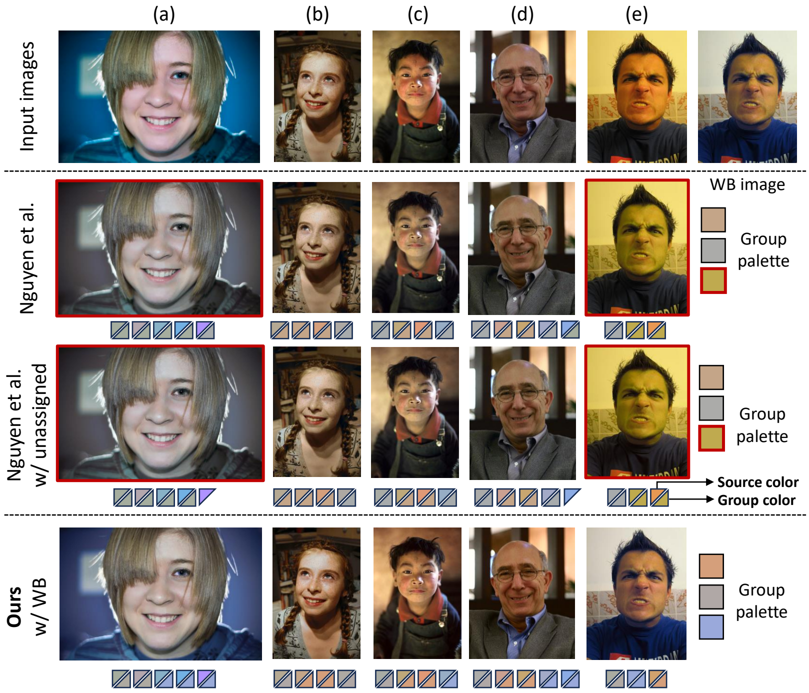


Figure 5: *Results after applying the white-balance correction module are presented in the following order: input images, results from two versions by Nguyen et al., and our results. In the images, red boxes are used to highlight unsatisfactory recolored images and palette colors. We obtain better color consistency among the recolored images by incorporating the white balance correction. The images used for this demonstration were sourced from Flickr [Fli04].*

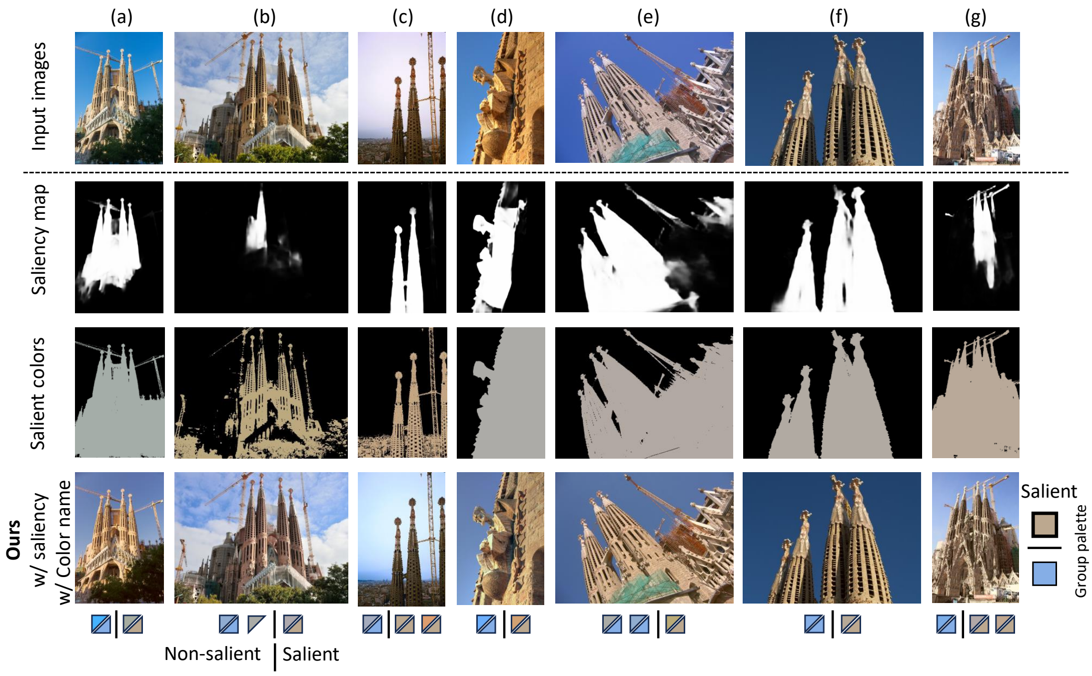


Figure 6: *Results of introducing saliency-guided palette grouping. Even when inaccurate saliency maps are used (second row), our saliency colors procedure (third row) can cover all the salient elements, producing recolored images (last row) with consistency in both the salient and the non-salient regions. Images are from Flickr [Fli04].*

Figure 7: *Example for our two approaches for saliency recoloring. From top to bottom: Original image, the two versions of Nguyen, the starting saliency map, the salient colors obtained by our procedure, and the results from our two versions. Red boxes mark the unsatisfactory recolored images. In our first result, we can see how the flowers are all converted to red by making both salient and non-salient regions consistent, and the leaves get a more middle green tone. In our second result, as we only match the non-salient regions, the flowers keep the same colors as in the original images, while the leaves are modified as in the previous case. We can see how we can obtain results that look natural in both of our results, in contrast to the results in Nguyen's approach. Images are from Flickr [Fli04].*

Color-naming association module. In Fig. 8, we show from top to bottom the input images, the results for both versions of Nguyen, and the results applying our color naming association module. As we can see in the figure, the standard module of Nguyen presents unrealistic colors in (a)-(c), while the unassigned version only solved the problem for (b), but it does not prevent the method from obtaining a greenish bird in (a) and blue clouds in (c). The reason is that, for this last version of Nguyen, the assignment is solved by minimizing the cost function giving a small penalization to the unassigned colors. However, this optimization does not consider any high-level perceptual features.

When applying our color naming association procedure, the results show how it addresses the previously mentioned problems. This is obtained by the ability of our module to break any matching in which the source palette and the group palette represent a different color name. Our method breaks a gray-green link in (a), a purple-blue link in (b), a gray-blue link in (c), and a brown-green link in (f).

Color naming helps mitigate drastic color changes (often due to hue shifts) in the recolored images. This restriction is well-suited for images containing objects with strong memory color associates, like skin tones, skies, and foliage/plants. Alternatively, relaxing the color name constraints or using no restrictions prioritizes larger consistency within the recolored collection, making it more suitable for creative applications like graphic design (Fig. 12). In such cases, applying color-naming association might limit the extent of achievable color transformations.

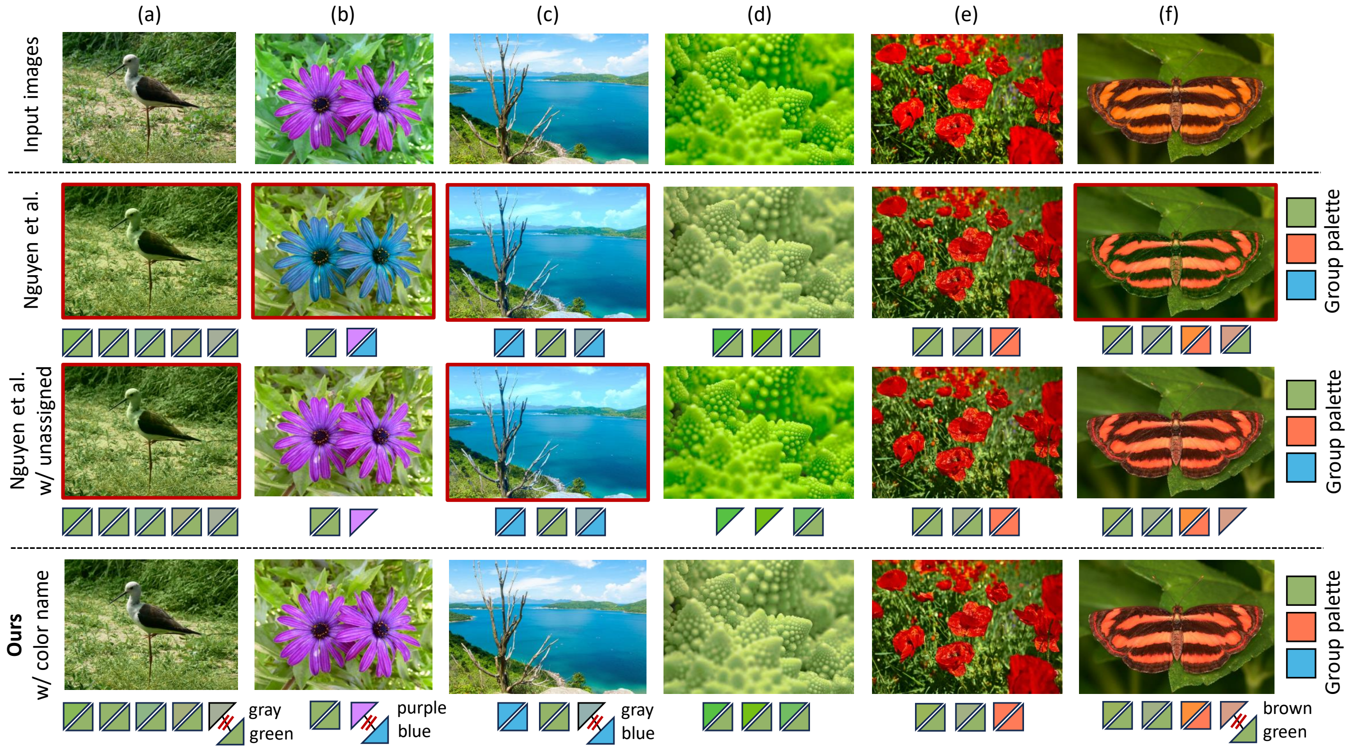


Figure 8: *Results of adding the color-naming association module. From top to bottom: Input images, the two versions by Nguyen et al., and our result. Red boxes highlight the unsatisfactory recolored images. In our results, we can see how the color-naming association module breaks matches in the (a), (b), (c), and (f), due to them having different color names in the source and the group palettes. This allows our method to obtain more natural images, avoiding the problems of the Nguyen et al. approach, namely in the green color of the bird in (a) and the blue color of the clouds in (c). Images are from Flickr [Fli04].*

Our color-naming association operates as a binary choice between source and target colors. Applying this association can compromise color consistency among the recolored collection in some specific scenarios. This said, our approach is still better than methods that directly interpolate between the source and group colors. Using direct interpolation might work for some cases (for example the flower image in Fig. 8 (b)) as "blue" and "purple" are similar enough colors; but direct interpolation drastically fails when is required to mix complementary colors, resulting in gray hues. An example of this is the red flower in Fig. 7 (a): the interpolation between "red" and "green" cannot produce a pleasant output.

Modules combination. We show the results of combining the saliency module with the other modules in Fig. 9. In the first two rows of the figure, the input images and the result for the unassigned version of Nguyen are shown. We can see how this last method presents washed-out colors in the t-shirts in (a) and how the yellow t-shirt and the yellow numbers in (c) are turned green.

The third and fourth rows show the saliency map and the salient colors used by our method. The last three rows show the results of using only our saliency module, both the saliency and the color naming modules, and all three modules. In Fig. 9 we use the saliency approach that matches both salient and non-salient regions.

We can see that using only the saliency module might not be enough, and some colors can still be changed in undesirable ways (see the white t-shirt in (d)). But combining the saliency module with the color-naming and the white-balance modules can reduce such problems, providing more color consistency images across the image collection.

4.3. User study

Given the subjective nature of our framework, we perform a user study to determine preferences among the different images. We created ten groups of images and computed the color consistency results by the two versions of Nguyen *et al.* [NPCB17] (standard and unassign) and three versions of our approach (only saliency, saliency + color naming, and saliency + color naming + white balance). The groups of images were selected to represent challenging scenarios for the baseline approach proposed in [NPCB17].

We also included in the experiment the input not-consistent images. Therefore, the number of comparisons was $10 \times 15 = 150$, where 15 is the number of combinations of 6 methods chosen in sets of 2.

The experiment consisted of a forced-choice pairwise paradigm, in which the groups of images obtained by any two of the methods were randomly shown on the left and right sides of the screen.

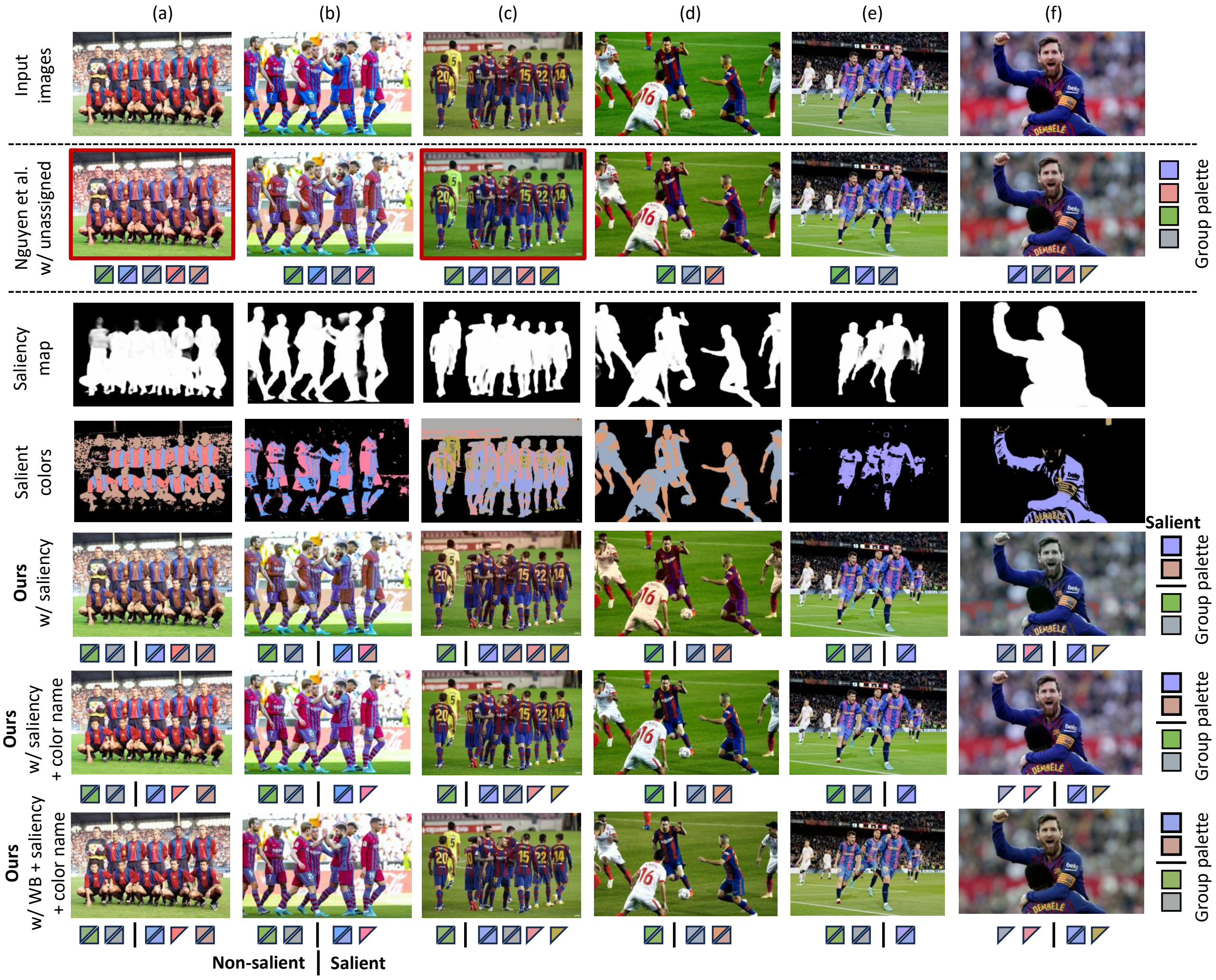


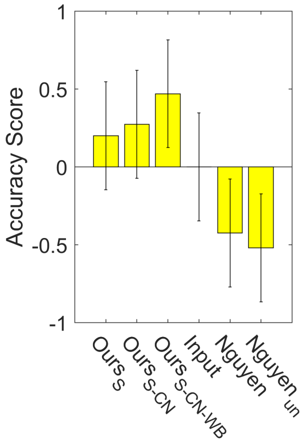
Figure 9: *Results for the combination of all our modules. From top to bottom we show the input images, the unassigned version of Nguyen, the starting saliency map, our salient colors, and our results with saliency, with saliency and color naming, and with all the three modules. Red boxes highlight the unsatisfactory recolored images. We can see how our approach is able to improve the results of Nguyen, especially in (a) and (c) -see the colors of the t-shirts in (a), and the yellow of the t-shirt and the numbers in (c)-. Images are from Flickr [Fli04].*

The experiment was conducted on a DELL P2317H monitor with the following x, y primaries—red: 0.6513, 0.3383; green: 0.3246, 0.6182; blue: 0.1556, 0.0441; white: 0.3114, 0.3328—with a peak white of 177.65 nits. The display was viewed at a distance of approximately 70 cm so that 40 pixels subtended 1 degree of visual angle. The experiment was conducted in a dark room.

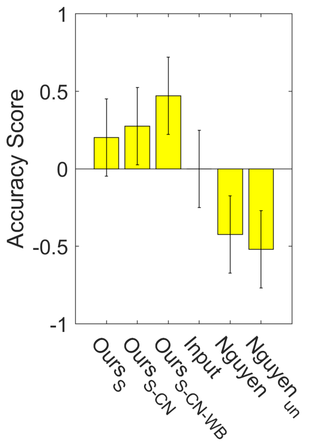
The study consisted of 15 observers. All observers had normal color vision (tested using the Ishihara color blindness test). The observers were asked to select the most color-consistent group of images while penalizing for both artifacts and unnatural colors.

We have analyzed the result of our experiment in terms of the Thurstone Case V Law of Comparative Judgment. Fig. 10 presents the results for the whole set of 150 comparisons. For readers unfamiliar with Thurstonian analysis [Thu27], a raw scoring matrix (that records the number of times each of the methods is preferred/not preferred against the others) is recorded. Various assumptions are made that allow the raw scores to be translated into a standardized (z-score) unit together with confidence intervals. The higher the z-score, the more a given algorithm is preferred.

In Fig. 10, we combine all the observers' results and convert the raw score matrix to the standardized z-score representation and the confidence intervals following the approaches of [Mor08] and [Mon06]. The average score is indicated by the yellow bars' top (or bottom). The vertical lines show the 95% confidence intervals. Clearly, Fig. 10 shows that our method delivers preferred outputs and, importantly, that both our method with saliency and color naming and our method with the three high-level features are statistically significantly better (at the 95% level) than the two versions from Nguyen *et al.* because the confidence intervals do not overlap.



(a) *with [Mor08]*



(b) *with [Mon06]*

Figure 10: *Results of the psycho-physical experiment using the Thurstone Case V test. S stands for saliency, CN for color naming, WB for white balance and "un" for unassigned. (a) and (b) shows the results of using two different methods to compute **confidence intervals**. Our method is statistically significantly better than the two versions of [NPCB17]. With (b) [Mon06] the statistical significance of our results stands with a larger difference.*

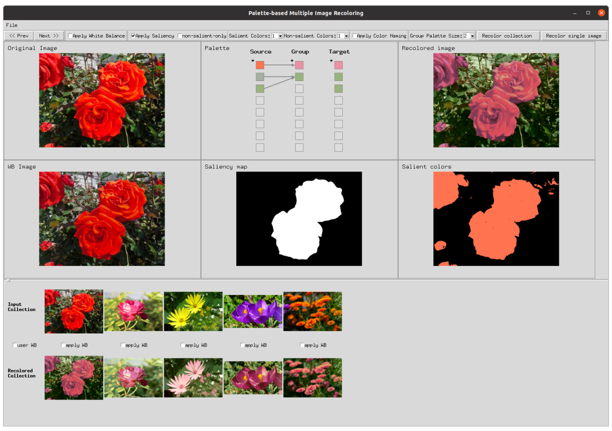


Figure 11: *Our GUI for interactive palette-based multi-image recoloring. See the supplementary material for a video recording of the GUI in use.*

4.4. Interactive multi-image recoloring

The framework presented in this paper is a fully working interactive system based on Python and Tkinter (see Fig. 11). Using our software, users can view and interact with the image recoloring process. Once the user loads a collection of images, the system automatically initiates the processing and provides visual representations of the source palette for each image, the group palette, and the resulting recolored image collection.

Our system allows users to choose between utilizing the modules described in the paper and manually selecting colors to manipulate the palettes. For a detailed demonstration of the interface, please refer to the supplemental materials, which include a screen recording.

4.5. Example of brochure design

Finally, Fig. 12 shows the ability of our method when used to prepare images for use in a brochure that uses an external color palette. We show in (a) the original brochure and in (b) and (c) two brochures in which the images have been modified by our framework using two different color palettes. In this example, the original images lack color consistency among themselves and with the brochure. Using our framework, the final brochure has a more consistent color appearance. In addition, Our method can adapt to different brochure color palettes. This example was computed by directly considering the given external color palette as the group palette of our framework.

5. Concluding remarks

We have introduced an interactive framework for achieving multi-image color consistency. Our framework incorporates white balance, saliency, and color naming within a general palette-based recoloring system. The combination of these additional high-level constraints significantly improves the overall results and produces recolored collections free from unwanted artifacts. Through qualitative examples and a user study, we have established that our approach surpasses the current state-of-the-art methods in terms of both visual quality and user preference.

We currently utilize saliency for its generalization ability and efficiency. As a binary classification, saliency detection forms the basis for incorporating complex semantic categories while maintaining computational efficiency. Future directions for this research will involve incorporating new modules that go one step further in prioritizing the semantic information within the images. One potential avenue is to leverage semantic segmentation methods [KMR*23, LWH*22] to identify and match similar semantic concepts. By integrating these techniques, we can enhance the recoloring process by considering the underlying meaning and context of the image content. This incorporation of semantic-based modules holds promise for further improving the overall quality and coherence of the recolored images.

Acknowledgements This paper was supported by Grant PID2021-128178OB-I00 funded by MCIN/AEI/10.13039/501100011033, ERDF "A way of making Europe", the Departament de Recerca i Universitats from Generalitat de Catalunya with reference 2021SGR01499 (JVC, LH, DX), the "Ayudas para la recualificación del sistema universitario español" financed by the European Union-NextGenerationEU (JVC), and Ramon y Cajal grant RYC2019-027020-I (LH), by National Natural Science Foundation of China under Grant No.U19B2037, by Natural Science Basic Research Program of Shaanxi Province Program No.2021JCW-03 (YZ). This work was also supported by the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) program, a NSERC Discovery Grant, and the Canada Research Chair program (MSB).

(a) Original brochure (b) Brochure with group palette 1 (c) Brochure with group palette 2

Figure 12: *Application of multi-image recoloring with different group palettes. (a) shows the original brochure. (b) and (c) show the brochures produced by images recolored with two different group palettes.*

References

[APCB19] Afifi M., Price B. L., Cohen S., Brown M. S.: When color constancy goes wrong: Correcting improperly white-balanced images. In *IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 1535–1544. 4

[BK91] Berlin B., Kay P.: *Basic color terms: Their universality and evolution*. Univ of California Press, 1991. 3, 5

[BPCD11] Bychkovsky V., Paris S., Chan E., Durand F.: Learning photographic global tonal adjustment with a database of input / output image pairs. In *IEEE Conference on Computer Vision and Pattern Recognition* (2011). 1, 6

[BT17] Barron J. T., Tsai Y.: Fast fourier color constancy. In *IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 6950–6958. 4

[Buc80] Buchsbaum G.: A spatial processor model for object colour perception. *Journal of the Franklin Institute 310*, 1 (1980), 1–26. 4

[BVB08] Benavente R., Vanrell M., Baldrich R.: Parametric fuzzy sets for automatic color naming. *JOSA A 25*, 10 (2008), 2582–2593. 2, 3

[CFL*15] Chang H., Fried O., Liu Y., DiVerdi S., Finkelstein A.: Palette-based photo recoloring. *ACM Trans. Graph. 34*, 4 (2015), 139:1–139:11. 2

[DLX*21] Du Z., Lei K., Xu K., Tan J., Gingold Y. I.: Video recoloring via spatial-temporal geometric palettes. *ACM Trans. Graph. 40*, 4 (2021), 150:1–150:16. 2

[Fai13] Fairchild M. D.: *Color appearance models*. John Wiley & Sons, 2013. 3

[Fli04] Flickr: Flickr. `https://www.flickr.com/`, 2004. 6, 7, 8, 9, 10

[GGvdW11] Gijsenij A., Gevers T., van de Weijer J.: Computational color constancy: Survey and experiments. *IEEE Trans. Image Process. 20*, 9 (2011), 2475–2489. 4

[HS12] Heer J., Stone M. C.: Color naming models for color selection, image editing and palette design. In *CHI Conference on Human Factors in Computing Systems* (2012), pp. 1007–1016. 3

[HSGL13] HaCohen Y., Shechtman E., Goldman D. B., Lischinski D.: Optimizing color consistency in photo collections. *ACM Trans. Graph. 32*, 4 (2013), 38:1–38:10. 1, 2

[HWL17] Hu Y., Wang B., Lin S.: Fc^4: Fully convolutional color constancy with confidence-weighted pooling. In *IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 330–339. 4

[IY18] Iwasa S., Yamaguchi Y.: Color selection and editing for palette-based photo recoloring. In *IEEE International Conference on Image Processing* (2018), pp. 2257–2261. 2, 3

[JXWS21] Jiang L., Xu M., Wang X., Sigal L.: Saliency-guided image translation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), pp. 16509–16518. 3

[KH18] Kang J., Hwang Y.: Hierarchical palette extraction based on local distinctiveness and cluster validation for image recoloring. In *IEEE International Conference on Image Processing* (2018), pp. 2252–2256. 3

[KML18] Kamkar S., Moghaddam H., Lashgari R.: Early visual processing of feature saliency tasks: A review of psychophysical experiments. *Frontiers in Systems Neuroscience 12* (10 2018), 54. 3

[KMR*23] Kirillov A., Mintun E., Ravi N., Mao H., Rolland C., Gustafson L., Xiao T., Whitehead S., Berg A. C., Lo W.-Y., Dollár P., Girshick R.: Segment anything. *arXiv:2304.02643* (2023). 11

[LPU*13] Lissner I., Preiss J., Urban P., Lichtenauer M. S., Zolliker P.: Image-difference prediction: From grayscale to color. *IEEE Trans. Image Process. 22*, 2 (2013), 435–446. 5

[LWH*22] Liu Q., Wen Y., Han J., Xu C., Xu H., Liang X.: Open-world semantic segmentation via contrasting and clustering vision-language embedding. In *European Conference on Computer Vision* (2022), pp. 275–292. 11

[LYC*15] Liu Y., Yuan Z., Chen B., Xue J., Zheng N.: Illumination robust color naming via label propagation. In *IEEE International Conference on Computer Vision* (2015), pp. 621–629. 3

[LZXD21] Liu X., Zhu L., Xu S., Du S.: Palette-based recoloring of natural images under different illumination. In *IEEE International*

Conference on Computer and Communication Systems, ICCCS (2021), pp. 347–351. 3

[MBK*23] MIANGOLEH S. M. H., BYLINSKII Z., KEE E., SHECHTMAN E., AKSOY Y.: Realistic saliency guided image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2023), pp. 186–194. 3

[MHJP18] MONROE W., HU J., JONG A., POTTS C.: Generating bilingual pragmatic color references. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT* (2018), pp. 2155–2165. 3

[Mon06] MONTAG E. D.: Empirical formula for creating error bars for the method of paired comparison. *J. Electronic Imaging 15*, 1 (2006), 010502. 10, 11

[Mor08] MOROVIČ J.: *Color gamut mapping*. John Wiley & Sons, 2008. 10, 11

[MSZ19] MECHREZ R., SHECHTMAN E., ZELNIK-MANOR L.: Saliency driven image manipulation. *Mach. Vis. Appl. 30*, 2 (2019), 189–202. 3

[NPCB17] NGUYEN H. M. R., PRICE B. L., COHEN S., BROWN M. S.: Group-theme recoloring for multi-image color consistency. *Comput. Graph. Forum 36*, 7 (2017), 83–92. 2, 3, 5, 6, 9, 11

[OHG12] OLKKONEN M., HANSEN T., GEGENFURTNER K. R.: High-level perceptual influences on color appearance. *Visual experience: Sensation, cognition, and constancy* (2012), 179–98. 3

[PTSK16] PARK J., TAI Y., SINHA S. N., KWEON I.: Efficient and robust color consistency for community photo collections. In *IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 430–438. 1, 2

[SSCO09] SHAPIRA L., SHAMIR A., COHEN-OR D.: Image appearance exploration by model-based navigation. In *Comput. Graph. Forum* (2009), vol. 28, pp. 629–638. 2

[TEG18a] TAN J., ECHEVARRIA J. I., GINGOLD Y. I.: Efficient palette-based decomposition and recoloring of images via rgbxy-space geometry. *ACM Trans. Graph. 37*, 6 (2018), 262. 2

[TEG18b] TAN J., ECHEVARRIA J. I., GINGOLD Y. I.: Palette-based image decomposition, harmonization, and color transfer. *CoRR abs/1804.01225* (2018). 1, 2

[Thu27] THURSTONE L. L.: A law of comparative judgment. *Psychological review 34*, 4 (1927), 273. 10

[TLG17] TAN J., LIEN J., GINGOLD Y. I.: Decomposing images into layers via rgb-space geometry. *ACM Trans. Graph. 36*, 1 (2017), 7:1–7:14. 2

[TNZ*20] TU Y., NIU L., ZHAO W., CHENG D., ZHANG L.: Image cropping with composition and saliency aware aesthetic score map. In *AAAI Conference on Artificial Intelligence* (2020), pp. 12104–12111. 3

[VCB17] VAZQUEZ-CORRAL J., BERTALMÍO M.: Gamut mapping for visual attention retargeting. In *Color and Imaging Conference* (2017). 3

[VCVBT12] VAZQUEZ-CORRAL J., VANRELL M., BALDRICH R., TOUS F.: Color constancy by category correlation. *IEEE Trans. Image Process. 21*, 4 (2012), 1997–2007. 4

[vdWSVL09] VAN DE WEIJER J., SCHMID C., VERBEEK J., LARLUS D.: Learning color names for real-world applications. *IEEE Trans. Image Process. 18*, 7 (2009), 1512–1523. 2, 5

[WLX19] WANG Y., LIU Y., XU K.: An improved geometric approach for palette-based image decomposition and recoloring. *Comput. Graph. Forum 38*, 7 (2019), 11–22. 2

[WWW*20] WEI J., WANG S., WU Z., SU C., HUANG Q., TIAN Q.: Label decoupling framework for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (2020), pp. 13022–13031. 5

[XYX*17] XIA M., YAO J., XIE R., ZHANG M., XIAO J.: Color consistency correction based on remapping optimization for image stitching. In *IEEE International Conference on Computer Vision Workshops* (2017), pp. 2977–2984. 2

[YCvdW18] YU L., CHENG Y., VAN DE WEIJER J.: Weakly supervised domain-specific color naming based on attention. In *International Conference on Pattern Recognition* (2018), pp. 3019–3024. 3

[YZvdW*18] YU L., ZHANG L., VAN DE WEIJER J., KHAN F. S., CHENG Y., PÁRRAGA C. A.: Beyond eleven color names for image understanding. *Mach. Vis. Appl. 29*, 2 (2018), 361–373. 3

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