Palette-based Color Transfer for Images and Videos

Yue Zhao
School of Mathematical Sciences
Xiamen University
zhaoyue99@stu.xmu.edu.cn

Zhonggui Chen
School of Informatics
Xiamen University
chenzhonggui@xmu.edu.cn

Juan Cao
School of Mathematical Sciences
Xiamen University
juancao@xmu.edu.cn

Abstract

In this paper, we introduce an enhanced color palette characterized by a heightened level of representativeness for images. This palette is not only easy to compute but also effectively conveys the distribution of pixels across the color space. Based on the proposed color palette, we present a tailored color palette matching algorithm designed for image color transfer by solving an optimization problem to transfer colors from a reference image to the original image. Our algorithm offers the flexibility to operate in fully automatic mode or provide various levels of user interactivity, allowing for coarse-to-fine editing. Moreover, we demonstrate the adaptability of our color palette and color transfer algorithm to diverse applications, including grayscale image colorization and temporally consistent, time-varying video color transfer. Extensive experiments and comparisons have been conducted to measure the quality of our results, employing both visual assessments and evaluation metrics. These findings demonstrate that our algorithm efficiently and faithfully transfers color styles from reference images to both color and grayscale images and videos.

Keywords: color palette, color transfer, image, video

1. Introduction

Color transfer involves adjusting the color attributes in an image or video to align with those found in a reference image or video. The central objective of color transfer is to preserve the inherent content and structure of the original while matching its color to that of the reference. This technique finds diverse applications across domains, including image processing [13], computer graphics [3], and multimedia [46]. For instance, it can be utilized to add color to grayscale images [4] and alter the ambiance of an image, such as transforming a daytime photo to a sunset scene [22].

The process of color transfer for images usually involves establishing a relationship between the color distributions of the original and reference images. This relation can be established through different methods, such as directly creating mappings between the color distributions of the two images [29, 14], or by employing manual interaction or image segmentation to establish color correspondences between them [4]. With the development of deep learning, neural networks are often used for color transfer and have made good progress [26, 17]. While these methods often demonstrate improved semantic alignment, they require a substantial amount of training data. Moreover, the outcomes obtained are not easily amenable to modification or adjustment, thereby limiting flexibility for subsequent editing. When directly extending image color transfer to video color transfer, both traditional and deep learning-based methods can encounter challenges, such as preserving color consistency across frames.

This paper introduces a novel image color transfer method that can be easily extended to video based on our generalized color palettes. Color palettes, referring to a collection made up of a limited number of colors, are considered as descriptors extracted from an image or video. Due to its simplicity, intuitiveness, versatility, and ease of computation, the color palette has been widely used in many fields in applications such as image color editing [7] or image labeling and retrieval [34]. However, color transfer through a color palette relies on manual color adjustments guided by the user's perception. Increasing the number of colors in the palette for complex images is necessary to convey color information, but it can make image editing more challenging for the user. To address this limitation, we introduce an innovative method for transferring colors from a user-specified reference image. We enhance color palettes to improve color representation and then automate the alignment of color palettes between the original and reference images. Importantly, this method is well-suited for extending to video content while ensuring temporal consistency, enabling dynamic color editing. The specific contributions of our work can be summarized as follows:

- We introduce an enhanced color palette with a high degree of representativeness for images, which effectively conveys the distribution of pixels across the color space and can be seamlessly extended to videos.
- We have developed an optimization-based algorithm

for color palette matching that seamlessly transfers colors from a reference image to a target image. Our algorithm provides flexibility by operating in fully automatic mode or by offering user interactivity at various levels, facilitating coarse-to-fine editing.

- Our proposed image color palettes and the corresponding color transfer algorithm can be easily extended to the applications of grayscale image colorization and time-varying or temporally consistent video color transfer and achieve faithful and efficient color transfer from reference to the input.

The paper is organized as follows: Section 2 briefly reviews several related works. Section 3 introduces the definition of our color palettes and its computation. Section 4 describes our image color transfer algorithm. In Section 5, we extend our image color transfer algorithm to grayscale image colorization and temporally consistent and time-varying color transfer for videos. Section 6 presents the experimental results and comparisons, and Section 7 concludes the paper.

2. Related work

In this section, we will provide a brief review of related works in the field of image and video editing, with a specific focus on palette-based editing and color transfer.

2.1. Palette-based editing

Palette-based image editing is an intuitive way for users to modify the color of an image. The user can edit the color of the image by modifying the colors of the palette. These color palette extraction methods can be categorized as follows:

Prominant colors-based color palettes. One potential approach to generating color palettes involves capturing the prominent colors from an image, which are those that can attract most of human attention. Various methods can be utilized to attain this objective and can be applied across a range of image-related applications. For example, Solli and Lenz [34] used the RGB histogram to export the color palette of the image for image labeling and retrieval. Siang Tan and Mat Isa [32] also started with the method of statistical analysis, using histogram thresholding to extract the color palette for image segmentation. Chang et al. [7] used RGB value clustering on image pixels to derive color palettes, and established a mapping function between the original and modified palettes, enabling color editing within images. Zhang et al. [47] obtained the palette's basis colors through image segmentation and represented the entire image's colors as linear combinations of these basis colors for image color editing. Aksoy et al. [2] presented a method to decompose an image into a set of soft color segments, which can be utilized for various image manipulation tasks. Zheng et al. [48] used Zhang et al. [47]'s color initialization method to select dominant colors and employed a color mapping technique to create a palette for categorical data visualization. Yan et al. [42] introduced a real-time lightweight convolutional network to quickly extract a certain number of color sets as an image palette for editing.

Polyhedron-based color palettes. Polyhedron-based color palettes employ geometric methods to extract palettes from the RGB color space of image pixels. Tan et al. [36] represented the color palette as the vertices of a simplified convex hull in RGB-space, mapping each pixel in the image to the target palette via nonlinear optimization. Subsequently, they introduced a straightforward method for automatically selecting the palette size [35]. However, convex-hull-based palettes may not effectively represent the dominant colors found in images. To address this limitation, Wang et al. [39] proposed an improved method for extracting a color palette that better represents the image's colors by relaxing the palette's geometry from a convex hull to a general polyhedron within the RGB color space. Additionally, Chao et al. [8] focused on expanding the usability of convex hull-based palettes (e.g., [36]) to support direct and localized image-space edits.

The above image color palettes cannot be directly applied to the video due to the need to consider temporal connections between adjacent frames. In [47], the authors extended image color palettes to videos by computing an average palette from sampled frames. However, in cases of substantial color variation among video frames, this average palette may lose some color information. In [9], the first palette-based video editing algorithm extended convex hull-based color palettes from images to spatial-temporal geometric palettes for videos. This allows users to modify palette colors for temporal variations. However, manual editing with palettes requires a small number of colors to avoid user inconvenience. It also relies on the user's ability to perceive colors, posing challenges for achieving desired color edits in complex images or videos.

2.2. Color transfer

Color transfer involves applying the color style of a reference image to the original image while preserving its content and structural information. We mainly focus on the color transfer algorithms for color images or videos. In this section, we review previous works by classifying them into traditional and deep learning-based methods.

Traditional methods. The pioneering algorithm by Reinhard et al. [31] matches the mean and standard deviation of the original and reference images. Pitie et al. [29] estimated a continuous transformation between their histograms, while Freedman and Kisilev [12] tackled the mapping of histogram distributions using the transport problem, computing the mean and variance of pixel values in each bin. An and Pellacini [3] achieved region correspondence between two images through user-stroke input, enabling color transfer between the corresponding regions. There are also methods for automatic image segmentation and region color matching, such as image segmentation based on texture information [4] and clustering-based algorithms [18]. Giraud et al. [13] proposed a fast superpixel-based color transfer method using a fast approximate nearest neighbor matching algorithm. Ferradans et al. [11] and Rabin et al. [30] solved the problem of color matching between two images with the optimal transport model. Grogan and Dahyot [14] represented the color distribution of the image with Gaussian Mixture model (GMM) and solved the color mapping function by minimizing the L_2 distance between the two distributions. Gu et al. [15] extended GMM to straightforwardly model the pixel-wise color distribution, and employ the Expectation-Maximization (EM) optimization to directly estimate the transferred pixel values.

Deep learning–based methods. In recent years, there have been several works using deep learning models to establish correspondence between images. Luan et al. [26] achieved photo-realistic transfer by constraining the transformation from the input to output to be locally affine in the color space and expressing this constraint as a custom fully differentiable energy term. He et al. [17] estimated a semantically dense correspondence between two images to achieve more accurate color transfer. Lee et al. [23] proposed a deep neural network for color transfer using histogram color analogy. Afifi et al. [1] proposed a color histogram-based method for controlling GAN-generated images' colors, encouraging the network to preserve the content of the original image while changing the color according to a given target histogram. Fang et al. [10] proposed a novel color transfer method based on the saliency information with brightness optimization.

Video color transfer. In contrast to image color transfer, video color transfer has received less attention, with current efforts primarily focusing on temporal consistency. Bonneel et al. [5] transferred colors between corresponding keyframes of the input and reference videos and then interpolated the transformations for temporal color consistency. However, due to the incorporation of image segmentation in the color transfer process, certain artifacts emerged in the outline. Yao et al. [43] used patch matching in the video to maintain temporal consistency, but their method had problems dealing with large moving objects in the scene. Considering luminance, color, and contrast, Zabaleta and Bertalmío [46] computed the color transformation from a selected video frame to the reference image, applying this transformation to ensure temporal consistency across all frames. Liu and Zhang [25] achieved temporally consistent color transfer by combining soft segmentation with GMM and quaternion. Yaosen et al. [44] generated a 3D lookup table for the video color transfer based on the neural network, so as to ensure a temporally consistent color transfer.

3. Color palette extraction and representation

In this section, we outline preliminary steps before introducing our color transfer algorithm. In this paper, all colors in the RGB color space have been normalized. We first describe a concise color palette that effectively represents the image's color information and can be easily extended to videos. Then we provide a brief introduction to generalized affine barycentric coordinates for image reconstruction.

3.1. Color palette extraction

Our color palette comprises a set of 3D points in the RGB color space. We construct the palette using the image's simplified RGB convex hull, incorporating dominant colors to improve its representativeness of the image.

Simplified RGB convex hull. We adopt the method proposed by Tan et al. [35] to extract the simplified RGB convex hull of the image. This method begins with the convex hull of all pixel colors represented as points in RGB space and iteratively simplifies it through edge contractions until a preset reconstruction error threshold is reached. The reconstruction error of the convex hull $\mathbf{H}$ with respect to the image $\mathbf{I}$ is defined as:

$$L(\mathbf{I},\mathbf{H}) = \sqrt{\frac{1}{|\mathbf{I}|}\sum_{\mathbf{p}\in\mathbf{I}} \|\mathbf{p}-\mathbf{p}_{\mathrm{H}}\|^2}, \tag{1}$$

where $\mathbf{p}$ represents the pixel color of the image $\mathbf{I}$, $|\mathbf{I}|$ denotes the number of pixels in $\mathbf{I}$, and $\mathbf{p}_{\mathrm{H}}$ is the point in the convex hull that is closest to $\mathbf{p}$. We can ensure that most pixel colors in the image are contained within the simplified convex hull by setting a lower threshold η for the reconstruction error. We assume that the number of vertices of the convex hull is N. These vertices (i.e., colors) will serve as the colors in our palette.

Dominant colors. To address the potential neglect of colors within the hull and avoid a loss of representative colors in the image, we augment the palette further by incorporating dominant colors extracted from the image. We employ the Canopy clustering algorithm [27] to group colors based on their similarity, initially yielding K clusters. Given that Canopy clustering may produce unstable results, we refine this by applying the improved K-means clustering method proposed by Chang et al. [7], using the number of clusters identified by Canopy clustering. The resulting K dominant colors are then incorporated into our palette.

"Clean-up" operation. We denote our initial palette as a collection of points in 3D color space: $\tilde{\mathbf{U}} = \{\mathbf{u}_i \in [0,1]^3 | i = 1,\cdots,N+K\}$. Directly combining the vertices of the convex hull with dominant colors may introduce redundancies. To eliminate redundancies, we conduct a "clean-up" process on the extracted initial palette $\tilde{\mathbf{U}}$. Specifically, for two colors $\{\mathbf{u}_i, \mathbf{u}_j\}$ in $\tilde{\mathbf{U}}$ that have the shortest Euclidean distance, we replace them with their average color $\frac{1}{2}(\mathbf{u}_i + \mathbf{u}_j)$. We repeat this clean-up operation on the updated palette until no color pair $\{\mathbf{u}_i, \mathbf{u}_j\}$ satisfies $||\mathbf{u}_i - \mathbf{u}_j|| < \varepsilon$, where ε is a predefined threshold. We denote the final color palette after "clean-up" as $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^n$, where n is the number of colors in the palette. An example of color palette extraction is illustrated in Fig. 1, where we have set the number of colors of the palette to be smaller for ease of demonstration. As depicted, the color of the convex hull vertices fails to capture the abundance of brown shades in the image, and the dominant colors compensate for this gap. In addition, our "clean-up" operation merges the two similar dark colors.

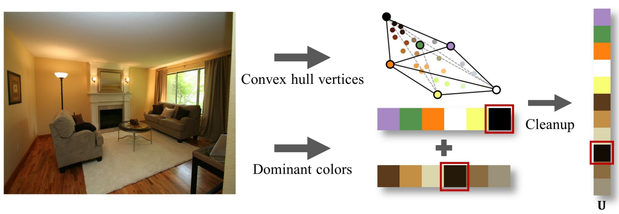


Figure 1: Our color palette of an image.

3.2. Color representation

We can well represent our image colors by using palette colors and affine generalized barycentric coordinates. Since our palette is defined as a set of points without topological relations in RGB space, we employ affine generalized barycentric coordinates [37], eliminating the need for connectivity information between points to represent the colors in the image. In particular, for an image $\mathbf{I}$ and its palette $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^n$, the affine generalized barycentric coordinates of any pixel color $\mathbf{p}$ with respect to $\mathbf{u}_i$ is given by

$$w(\mathbf{u}_i, \mathbf{p}) = \langle \mathbf{p} - \mathbf{c}, (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{x}_i \rangle + \frac{1}{n}, \tag{2}$$

where $\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{u}_i$ is average color of $\mathbf{U}$, $\mathbf{x}_i = \mathbf{u}_i - \mathbf{c}$, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$, and $\langle \cdot, \cdot \rangle$ denotes the inner product. Then, the pixel color $\mathbf{p}$ can be represented as a weighted linear combination of all colors in $\mathbf{U}$ as follows:

$$\mathbf{p} = \sum_{i=1}^{n} w(\mathbf{u}_i, \mathbf{p})\mathbf{u}_i, \forall \mathbf{p} \in \mathbf{I}. \tag{3}$$

It is worth noting that, according to Eqn. (2), since $(\mathbf{X}\mathbf{X}^T)^{-1}$ is a positive definite matrix, the coordinate $w(\mathbf{u}_i, \mathbf{p})$ may take on negative values when $\langle \mathbf{p} - \mathbf{c}, \mathbf{x}_i \rangle < 0$. However, the presence of negative values in the affine generalized barycentric coordinates does not affect our recoloring process; further analysis is provided in Section 6.2.

4. Image color transfer

By specifying a reference image, our algorithm enables automatic color transfer. Our algorithm also allows users to manually select areas and directly edit the color palette for fine-tuning. This enhances users' flexibility and provides a broader array of alternatives in color editing. Examples of coarse-to-fine editing are shown in Fig. 2.

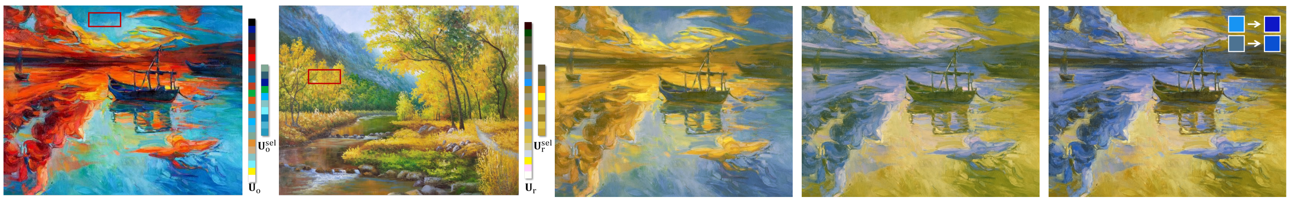
4.1. Reference image-based color transfer

Given an original image $\mathbf{I}_{\text{o}}$ and a reference image $\mathbf{I}_{\text{r}}$, our image color transfer algorithm aims to replicate the color effects of $\mathbf{I}_{\text{r}}$ onto $\mathbf{I}_{\text{o}}$. We extract the initial color palette for both the original image $\mathbf{I}_{\text{o}}$ and the reference image $\mathbf{I}_{\text{r}}$, denoted as $\tilde{\mathbf{U}}_{\text{o}}$ and $\tilde{\mathbf{U}}_{\text{r}}$, respectively. After performing "clean-up" operations on each, we obtain the final palettes $\mathbf{U}_{\text{o}}$ and $\mathbf{U}_{\text{r}}$, with the respective color counts n_{o} and n_{r}. In this section, we automatically match colors between palettes $\mathbf{U}_{\text{o}}$ and $\mathbf{U}_{\text{r}}$ by solving a tailored constrained optimization problem.

We first define non-negative variables $T_{ij} \in [0, 1]$, for $i = 1, \ldots, n_{\text{o}}$ and $j = 1, \ldots, n_{\text{r}}$, to indicate the matching relationships between the colors in $\mathbf{U}_{\text{o}}$ and $\mathbf{U}_{\text{r}}$. A value of $T_{ij} = 1$ indicates a one-to-one match between the i-th color $\mathbf{u}_i^{\text{o}}$ in $\mathbf{U}_{\text{o}}$ and the j-th color $\mathbf{u}_j^{\text{r}}$ in $\mathbf{U}_{\text{r}}$. A color in $\mathbf{U}_{\text{o}}$ may match multiple colors in $\mathbf{U}_{\text{r}}$. In such cases, we require the sum $\sum_{j=1}^{n_{\text{r}}} T_{ij}$ equals 1, indicating that $\mathbf{u}_i^{\text{o}}$ corresponds to a weighted combination of colors $\sum_{j=1}^{n_{\text{r}}} T_{ij}\mathbf{u}_j^{\text{r}}$. Additionally, to ensure that each color in the palette $\mathbf{U}_{\text{r}}$ is used as extensively as possible for recoloring, and to avoid the situation where a color is hardly used, we want the value of $\sum_{i=1}^{n_{\text{o}}} T_{ij}$ to be as large as possible for each j. According to the constraint $\sum_{j=1}^{n_{\text{r}}} T_{ij} = 1$, the maximum value of $\sum_{i=1}^{n_{\text{o}}} T_{ij}$ is $\frac{n_{\text{o}}}{n_{\text{r}}}$. Therefore, we set the constraint $\sum_{i=1}^{n_{\text{o}}} T_{ij} = \frac{n_{\text{o}}}{n_{\text{r}}}$. Then, we solve the constrained optimization problem for T_{ij}, where $i = 1, \ldots, n_{\text{o}}$ and $j = 1, \ldots, n_{\text{r}}$, as follows:

$$\begin{aligned} &\min_{T_{ij}} \{E_{\text{sim}} + \beta E_{\text{brt}}\}, \\ &s.t. \sum_{i=1}^{n_{\text{o}}} T_{ij} = \frac{n_{\text{o}}}{n_{\text{r}}}, j = 1, \ldots, n_{\text{r}}, \\ &\quad \sum_{j=1}^{n_{\text{r}}} T_{ij} = 1, i = 1, \ldots, n_{\text{o}}, \\ &\quad T_{ij} \geq 0, i = 1, \ldots, n_{\text{o}}, j = 1, \ldots, n_{\text{r}}, \end{aligned} \tag{4}$$

where E_{sim} and E_{brt} are two terms, described in detail later, used to encourage matching between similar colors, achieve comparable average brightness and variance to the reference image, and balance these attributes, respectively. The parameters β are used to balance these terms.



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

Figure 2: Color palettes and results of automatic/interactive color transfer. (a) Input image; (b) reference image; (c) automatic result; (d) result with prioritized matching of the color of the two regions indicated by red boxes; and (e) result with two colors from the palette updated. The color palettes for the entire images and the selected regions are displayed separately on the sides.

The *color similarity term* E_{sim} aims to match similar colors between the two palettes as effectively as possible, which is a fundamental principle of our palette matching approach. This term is designed to encourage the recolored image to imitate a color style similar to that of the reference image and is defined as follows:

$$E_{\text{sim}} = \sum_{i=1}^{n_{\text{o}}} \sum_{j=1}^{n_{\text{r}}} T_{ij} \left\| \mathbf{u}_i^{\text{o}} - \mathbf{u}_j^{\text{r}} \right\|_2, \ \mathbf{u}_i^{\text{o}} \in \mathbf{U}_{\text{o}}, \ \mathbf{u}_j^{\text{r}} \in \mathbf{U}_{\text{r}}. \quad (5)$$

While E_{sim} focuses on color similarity, it does not ensure that the brightness of the recolored image remains within a reasonable range. Therefore, the *brightness constraint term* E_{brt} is used to constrain the distribution of brightness in the recolored image. This term is defined as:

$$E_{\text{brt}} = |\text{Ave}(\mathbf{I}_{\text{o}}') - \text{Ave}(\mathbf{I}_{\text{r}})| + |\text{Var}(\mathbf{I}_{\text{o}}') - \text{Var}(\mathbf{I}_{\text{r}})|, \quad (6)$$

where $\text{Ave}(\cdot)$ and $\text{Var}(\cdot)$ denote the mean and variance of brightness for all pixels in the image, respectively. The first part of E_{brt} is the difference between average brightness of the reference image $\mathbf{I}_{\text{r}}$ and the recolored image $\mathbf{I}_{\text{o}}'$, aiming to align the overall brightness of $\mathbf{I}_{\text{r}}$ as closely as possible with that of $\mathbf{I}_{\text{o}}'$. Considering only the average brightness difference between two images may cause the recolored image's pixel brightness to average out. And the second part of E_{brt}, which is the difference between brightness variance of the two images, can alleviate this undesirable phenomenon.

We defer the discussion regarding the impact of the weights β and the effectiveness of the terms E_{brt} to Section 6.1. We solve this constrained optimization problem using the Cplex solver [20]. Once the values of T_{ij} are determined, we replace the colors $\mathbf{u}_i^{\text{o}}$ in Eqn. (3) with their corresponding weighted combinations $\sum_{j=1}^{n_{\text{r}}} T_{ij} \mathbf{u}_j^{\text{r}}$, while retaining the original mixing weights. Namely, each pixel originally colored $\mathbf{p}$ is recolored with the new color $\mathbf{p}'$ as follows:

$$\mathbf{p}' = \sum_{i=1}^{n_{\text{o}}} w(\mathbf{u}_i^{\text{o}}, \mathbf{p}) \sum_{j=1}^{n_{\text{r}}} T_{ij} \mathbf{u}_j^{\text{r}}, \forall \mathbf{p} \in \mathbf{I}_{\text{o}}. \quad (7)$$

Consequently, we recolor each pixel of the original image $\mathbf{I}_{\text{o}}$ to obtain the color-transferred result $\mathbf{I}_{\text{o}}'$. This image retains the original structure while adopting the color effects of the reference image $\mathbf{I}_{\text{r}}$; see Fig. 2(a-c).

4.2. Interaction-based color transfer

We further enhance the user experience by allowing users to specify areas of the images and match their colors to achieve more detailed color adjustments; see Fig. 2(d). Our method also enables users to edit image colors by directly modifying the palette; see Fig. 2(e).

Local color palette extraction. Suppose the user selects two regions, $\mathbf{S}_{\text{o}}$ and $\mathbf{S}_{\text{r}}$, from the original and reference images, respectively. We prioritize the color correspondence within these specified regions. To achieve this goal, we construct local color palettes $\mathbf{U}_{\text{o}}^{\text{sel}}$ and $\mathbf{U}_{\text{r}}^{\text{sel}}$ for the selected regions from the color palettes $\mathbf{U}_{\text{o}}$ and $\mathbf{U}_{\text{r}}$. Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$ be two vectors. We introduce an operator such that $\mathbf{u} \preceq \mathbf{v}$ denotes that the absolute value of each element in $\mathbf{u}$ is smaller than the corresponding element in $\mathbf{v}$. Then, we set

$$\mathbf{U}_{\text{o}}^{\text{sel}} = \{\mathbf{u}_i^{\text{o}} | \ \text{HSV}(\mathbf{u}_i^{\text{o}}) - \text{HSV}(\bar{\mathbf{S}}_{\text{o}}) \preceq \mathbf{a}, \mathbf{u}_i^{\text{o}} \in \mathbf{U}_{\text{o}}\}, \quad (8)$$

$$\mathbf{U}_{\text{r}}^{\text{sel}} = \{\mathbf{u}_i^{\text{r}} | \ \text{HSV}(\mathbf{u}_i^{\text{r}}) - \text{HSV}(\bar{\mathbf{S}}_{\text{r}}) \preceq \mathbf{a}, \mathbf{u}_i^{\text{r}} \in \mathbf{U}_{\text{r}}\}, \quad (9)$$

where $\bar{\mathbf{S}}_{\text{o}}$ and $\bar{\mathbf{S}}_{\text{r}}$ are the average colors of the pixels in the $\mathbf{S}_{\text{o}}$ and $\mathbf{S}_{\text{r}}$, respectively, $\text{HSV}(\cdot)$ converts color from the RGB space to the HSV space, and $\mathbf{a} \in \mathbb{R}^3$ is a predefined threshold vector. Intuitively, palettes for selected regions include colors from the entire palette that resemble the colors and hues in chosen areas. Note that $\mathbf{U}_{\text{o}}^{\text{sel}}$ and $\mathbf{U}_{\text{r}}^{\text{sel}}$ may consist of different numbers of colors, denoted as $n_{\text{o}}^{\text{sel}}$ and $n_{\text{r}}^{\text{sel}}$, respectively. We sort the colors in $\mathbf{U}_{\text{o}}^{\text{sel}}$ (resp. $\mathbf{U}_{\text{r}}^{\text{sel}}$) in ascending order according to their Euclidean distance to $\bar{\mathbf{S}}_{\text{o}}$ (resp. $\bar{\mathbf{S}}_{\text{r}}$).

Local color palette matching. First, let's consider the color matching between $\mathbf{U}_{\text{o}}^{\text{sel}}$ and $\mathbf{U}_{\text{r}}^{\text{sel}}$. If $n_{\text{o}}^{\text{sel}} > n_{\text{r}}^{\text{sel}}$, we divide the sorted colors in $\mathbf{U}_{\text{o}}^{\text{sel}}$ into $n_{\text{r}}^{\text{sel}}$ parts as evenly as possible and then match the corresponding colors in $\mathbf{U}_{\text{r}}^{\text{sel}}$. If $n_{\text{o}}^{\text{sel}} \leq n_{\text{r}}^{\text{sel}}$, we sequentially match the colors in $\mathbf{U}_{\text{o}}^{\text{sel}}$ to the first $n_{\text{o}}^{\text{sel}}$ colors in $\mathbf{U}_{\text{r}}^{\text{sel}}$ in order. Next, we consider

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

Figure 3: An example of our grayscale image colorization. (a) Grayscale image, (b) reference image, (c) result without transition region processing, and (d) result with transition region processing.

the matching of the remaining colors in the palettes. We incorporate the established matching relationships between selected colors $\mathbf{U}_{\mathrm{o}}^{\mathrm{sel}}$ and $\mathbf{U}_{\mathrm{r}}^{\mathrm{sel}}$ as hard constraints into the optimization problem in Eqn. (4). For instance, if $n_{\mathrm{o}}^{\mathrm{sel}} > n_{\mathrm{r}}^{\mathrm{sel}}$ and $\{\mathbf{u}_{i_k}^{\mathrm{o}}\}_{k=1}^{m} \subset \mathbf{U}_{\mathrm{o}}^{sel}$ matches the color $\mathbf{u}_{j_l}^{\mathrm{r}}$ in $\mathbf{U}_{\mathrm{r}}^{sel}$, then we add the following constraint:

$$\sum_{k=1}^{m} T_{i_k j_l} = \frac{n_{\mathrm{o}}}{n_{\mathrm{r}}}. \tag{10}$$

By solving Eqn. (4) with the new constraints added, we can determine the matching relationships for all colors in the palettes. So far, each color in the original image's palette corresponds to some colors in the reference image's palette. We can then recolor our image using Eqn. (7).

5. Applications to grayscale images and videos

Our image color palette extraction and matching method can easily be extended to grayscale image colorization and video color transfer. In this section, we describe the details of these applications.

5.1. Grayscale image colorization

For effective color transfer from a colorful reference image $\mathbf{I}_{\mathrm{r}}$ to a grayscale image $\mathbf{I}_{\mathrm{o}}$, we utilize the matting network [24] to extract foreground and background, creating a mask with values ranging in $[0, 1]$, where 1 denotes foreground, 0 denotes background, and in-between values indicate ambiguous ownership; see the upper right corner of Fig. 3(a&b). We colorize the foreground and background separately. We extract the foreground and background palettes using the method outlined in Section 3.1, setting $N = 0$. In order to enhance the informativeness of the extracted palette, we take into account the pixel positions on the image when extracting the palette of a grayscale image. In particular, we represent each pixel $\mathbf{p}$ in $\mathbf{I}_{\mathrm{o}}$ with a 3D point $(p_{\mathrm{g}}, \omega x, \omega y)$. Here, p_{g} denotes its gray value, (x, y) represents the pixel's position in the image, and ω is a parameter experimentally set to be 0.2. We establish an optimization problem between two foreground palettes for automatic color matching, where the energy function focuses solely on E_{sim} and substitutes color differences with brightness differences between two pixels. The same process is then applied to the background. Finally, to achieve a smooth foreground contour in the result, the color of $\mathbf{p}$ is determined as:

$$\mathbf{p} = p_{\mathrm{mask}} \mathbf{c}_{\mathrm{f}} + (1 - p_{\mathrm{mask}})\, \mathbf{c}_{\mathrm{b}},$$

where $p_{\mathrm{mask}} \in [0, 1]$ is the corresponding value in the mask, $\mathbf{c}_{\mathrm{f}}$ and $\mathbf{c}_{\mathrm{b}}$ are the color reconstructed using the foreground palette and the background palette. The comparison results in Fig. 3(c&d) validate the effectiveness of our transition region processing.

5.2. Video color transfer

We extend the single-image color transfer framework to video color editing. Similar to the algorithm on images, our video color transfer is also example-based. We propose two video editing modes: temporally consistent color transfer and time-varying color transfer.

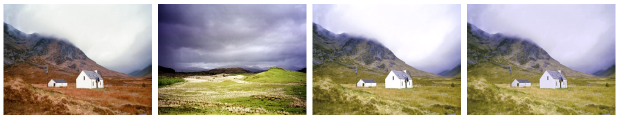
5.2.1 Temporally consistent color transfer

Video frames may exhibit color variations due to scene or lighting changes. Temporally consistent color transfer seeks to maintain color consistency across frames, ensuring uniform editing effects throughout the video, applicable to tasks such as movie color grading [5]. To ensure temporally consistent color transfer in videos, a common approach is to apply a shared color palette for all frames, as demonstrated in [47]. Without specific geometric requirements for our palette colors, we simply choose the one with the largest variance among all frame palettes as the common color palette. In videos with continuous scene changes, our extended approach maintains consistent colors across all frames, yielding satisfactory color transfer results, as discussed in Section 6.4 and illustrated in Figs. 13&14 and videos in the Supplementary Materials.

5.2.2 Time-varying color transfer

Time-varying color transfer enables color effects that change over time between adjacent frames, which can bring richer and more interesting edits [9]. Our time-varying color transfer enables users to specify reference images for any frame, achieving color transfer and ensuring continuity in color changes between edited frames.

First, we extract the keyframe sequence from the video frame set $\{\mathbf{F}_i\}$ using the inter-frame difference method [33], which provides a concise representation of the content. Given the similar color distribution between adjacent keyframes $\mathbf{F}_p$ and $\mathbf{F}_q$ (where $q > p$), we set the color palette for the intermediate frames $\mathbf{F}_{p+k}$ (for



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

Figure 4: Comparison of the results before and after adding the energy terms E_{brt}. (a) Original image, (b) reference image, (c) result without E_{brt}, and (d) result with E_{brt}.

$k = 1, \ldots, q-p-1$) to match that of keyframe $\mathbf{F}_p$. Specifically, we extract the initial color palette for $\mathbf{F}_p$ as described in Section 3.1. The intermediate frames $\mathbf{F}_{p+k}$ are assigned palettes with the same number of colors from the simplified convex hull and dominant colors as in the keyframe $\mathbf{F}_p$. Finally, a "Clean-up" operation is applied to $\tilde{\mathbf{U}}_{p+k}$ (for $k = 0, \ldots, q-p-1$) until no further adjustments are needed, with the final palette for the i-th frame denoted as $\mathbf{U}_i$.

For any two adjacent keyframes $\mathbf{F}_p$ and $\mathbf{F}_q$ (where $q > p$), we establish correspondence between the palettes $\mathbf{U}_p$ and $\mathbf{U}_{p+k}$, $k = 1, 2, \ldots, q-p$, by solely minimizing the color similarity term E_{sim}. Assume that the user specifies a reference image for the frame $\mathbf{F}_j$ between $\mathbf{F}_p$ and $\mathbf{F}_q$. We obtain the updated color palette $\mathbf{U}'_j$ through automatic color transfer. Palettes for other frames are computed by using linear interpolation between the color palettes of $\mathbf{U}_p$ ($\mathbf{U}_q$) and $\mathbf{U}'_j$. This process updates the palettes for frames between keyframes $\mathbf{F}_p$ and $\mathbf{F}_q$. Finally, We recolor the corresponding frames using Eqn. (7); see Fig. 15.

6. Experiments

In this section, we present experimental results for image and video color transfer, along with grayscale image colorization. We compare our algorithm against traditional and state-of-the-art methods to demonstrate its efficacy. The experiments were conducted on a PC with a 2.6 GHz CPU and 16 GB RAM. In our experiments, without additional explanation, we set parameters in our algorithm to the following default values: $\eta = 0.001, \varepsilon = 0.1$ in Section 3.1, and $\mathbf{a} = [0.2, 0.4, 0.3]^T$ in Section 4.2.

6.1. Ablation experiment on the *brightness constraint term* E_{brt}

In this section, we discuss the energy term E_{brt} of the optimization problem presented in Section 4.1. We demonstrate the effectiveness of the *brightness constraint term* E_{brt} in Fig. 4. Since we perform non-interactive color transfer, we focus here solely on the overall color effect of the recolored results. As shown in Fig. 4, the colors obtained from solving the optimization problem with complete energy terms are more in line with the contrast of the reference image. Next, we will discuss the weight β with respect to E_{brt}.

When establishing the optimization problem, we aim to satisfy the matching between similar colors as much as possible, with the *brightness constraint term* serving merely as auxiliary to further optimize the brightness of the results. By the definition of E_{sim} and E_{brt}, the value of E_{sim} is roughly n_{o} times that of E_{brt}. In order to make the term E_{sim} the primary focus of the optimization, the weight β of E_{brt} cannot be too large. Otherwise, when solving the optimization problem, the principle of matching similar colors will be violated in an attempt to reduce the difference of the mean and variance of brightness between images, as an example shown in Fig. 5. When β is set to n_{o}, the blue in the original palette starts to match the black in the reference palette, which is not the result we expect. And it has been found through multiple experiments that setting β to around $0.4n_{\text{o}}$ generally produces stable and robust results.

6.2. Results of image color transfer

The color palettes in this paper integrate convex hull-based representation in RGB space with dominant colors extracted from the image. Fig. 6 demonstrates that relying solely on either representation fails to faithfully transfer the color style. Our example-based method automatically transfers colors and eliminates the need for tedious manual editing. If users are unsatisfied with the automatically generated results, they can achieve more tailored outcomes through interaction, as described in Section 4.2 and shown in Fig. 2(c&d). We present additional interaction-based color transfer results in Fig. 7.

Comparison with traditional methods. The optimal transport-based methods [11, 30] primarily employ optimal transport between color distributions, resulting in artifacts when there is a significant distribution difference between two images. In contrast, our method produces a more natural and artifact-free outcome; see Fig. 8. In Fig. 9, the statistics-based method [31] and the color distribution-based method [29] struggle to effectively transfer the color information from the reference image to the original image, resulting in noticeable discrepancies between the transferred and target colors. The superpixel-based method [13] may also falter in representing the dominant color of the reference image, as evidenced in the flower's color in the first row of Fig. 9. Grogan and Dahyot [14] employs GMM to represent the color distribution, leading to artifacts along object boundaries, such as the contour of the flower.

Comparison with deep learning-based methods. We compare our method with three state-of-the-art deep learning-based methods [26, 17, 44, 10] in Fig. 10. These deep learning-based methods fail to transfer colors effectively between images with no obvious semantic correspondence. As demonstrated in the second row of Fig. 10, where the input and reference images differ significantly in seman-

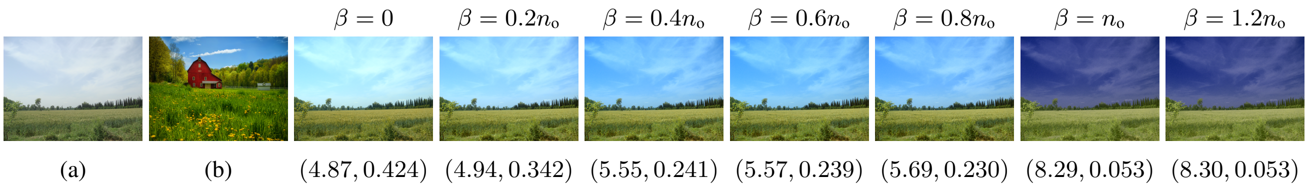


Figure 5: Comparison of results under different values of β. (a) Original image; (b) reference image; and $(\cdot, \cdot)$ represents the values of E_{sim} and E_{brt}, respectively.

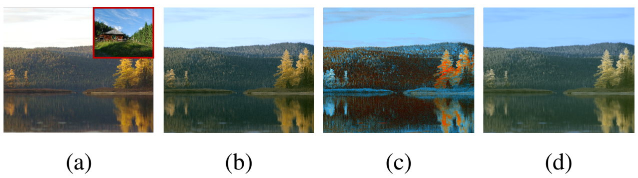


Figure 6: Comparison of color transfer results with different color palettes. (a) Original image and reference image shown in the right-top corner; (b) recolored image with convex hull vertices; (c) recolored image with dominant colors; and (d) recolored image with our palette.

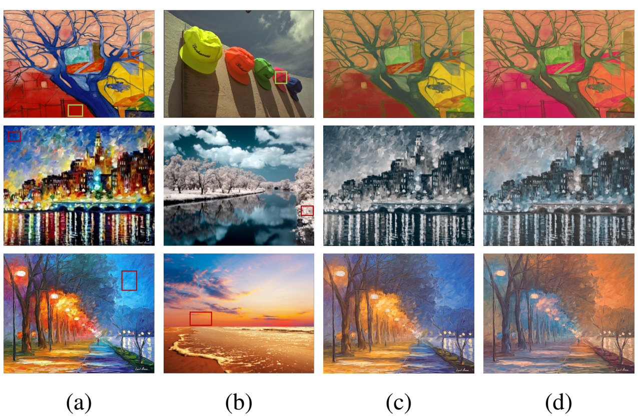


Figure 7: Results of interaction-based image color transfer. (a) Original image; (b) reference image; (c) automatic color transfer results; (d) interactive color transfer results.

tic content, the method in [26] introduces artifacts across the entire image, while the results by [17] and [44] display an unnatural halo around the tree on the left side of the image. Additionally, Fang et al. [10] performed color transfer based on saliency detection. However, when the salient areas of the image are unclear, this approach can lead directly to failures in color transfer. Even when the input and reference images exhibit better semantic similarity, artifacts such as un-transferred regions, missed structures, and unnatural colors still occur, as shown in the buildings in the fifth row of Fig. 10. While our method cannot guarantee color transfer between semantically corresponding objects,

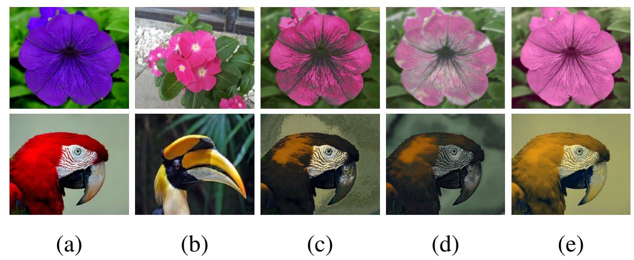


Figure 8: Comparisons with optimal transport-based color transfer methods. (a) Original image; (b) reference image; (c) Ferradans et al. [11]; (d) Rabin and Papadakis [30]; and (e) ours.

it consistently produces more natural results compared to deep learning-based methods.

Quantitative comparisons. We employ the structure similarity index measure (SSIM) [40] as our quantitative metric to measure structural similarity between the original and recolored images. It indicates the extent of artifacts introduced during color transfer, with values ranging from 0 to 1—higher values denote greater structural similarity. Table 1 compares SSIM values for recoloring results in Figs. 9&10 between traditional and deep learning-based methods. Our method notably achieves a higher SSIM in most cases, indicating fewer artifacts and more stable result quality. Note that higher SSIM values, as demonstrated by Pitie et al. [29] in Table 1(a), do not necessarily ensure visually better transfer results. For instance, as shown in Fig. 9, they did not successfully transfer the color of the reference image to the original image in some examples, such as the tree in the third row.

User study. To assess the color transfer results' quality more effectively, we conducted a user study, evaluating our method and others in terms of perceptual realism and faithfulness to reference color. Perceptual realism refers to the absence of unnatural colors, artifacts, noise and halos in the image, while faithfulness to reference color refers to the similarity between the color style of the resulting image and the reference image. We use 7 different examples for each of the four traditional methods and four deep learning-based methods, and collect the responses from 30 participants.

(a) (b) (c) (d) (e) (f) (g)

Figure 9: Comparisons with traditional color transfer methods. (a) Original image; (b) reference image; (c) Reinhard et al. [31]; (d) Pitie et al. [29]; (e) Giraud et al. [13]; (f) Grogan and Dahyot [14]; and (g) ours.

SSIM	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6
[31]	**0.69**	0.81	0.96	0.97	0.97	0.99
[29]	0.56	0.81	**0.97**	0.98	0.97	**0.99**
[13]	0.43	0.73	0.94	0.96	0.92	0.89
[14]	0.68	0.59	0.89	0.96	0.78	0.95
Ours	0.50	**0.91**	0.96	**0.99**	**0.99**	0.98

(a)

SSIM	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6
[26]	**0.56**	0.31	0.67	0.22	0.34	0.77
[17]	0.55	0.70	0.86	0.28	0.60	0.70
[44]	0.52	0.83	0.73	**0.80**	0.56	**0.96**
[10]	0.26	0.32	0.72	0.39	0.94	0.96
Ours	0.52	**0.99**	**0.91**	0.65	**0.99**	0.92

(b)

Table 1: Comparison of SSIM on color transfer results. (a) Comparison with traditional methods. (b) Comparison with the methods based on deep learning.

We randomly showed participants the results of different approaches and asked them to rate the results on the two aspects. The rating scale ranged from 1 to 5, with higher scores for results that are more consistent with perceptual realism or more faithful to the color of the reference image. We give the average and standard deviation of the scores for each method in Table 2. Our method has a larger average score and a smaller standard deviation than both classes of methods. We also asked participants to choose the results they thought were best in terms of perceptual realism and faithfulness to reference color, and the column C.B. in Table 2 shows the percentage of each method chosen as best. The results indicate a preference for our method over both traditional and deep learning-based methods.

The effect of negative coordinates on recoloring. In our calculations of affine generalized barycentric coordinates, negative values may arise. When these coordinates are utilized for recoloring with the reference palette, there is a potential risk of generating colors that fall outside the convex hull of the reference image, which could compromise the quality of the color transfer. To assess this impact, we employ Eqn. (1) to compute the reconstruction error between the simplified convex hull of the reference image and the resulting recolored image across all examples

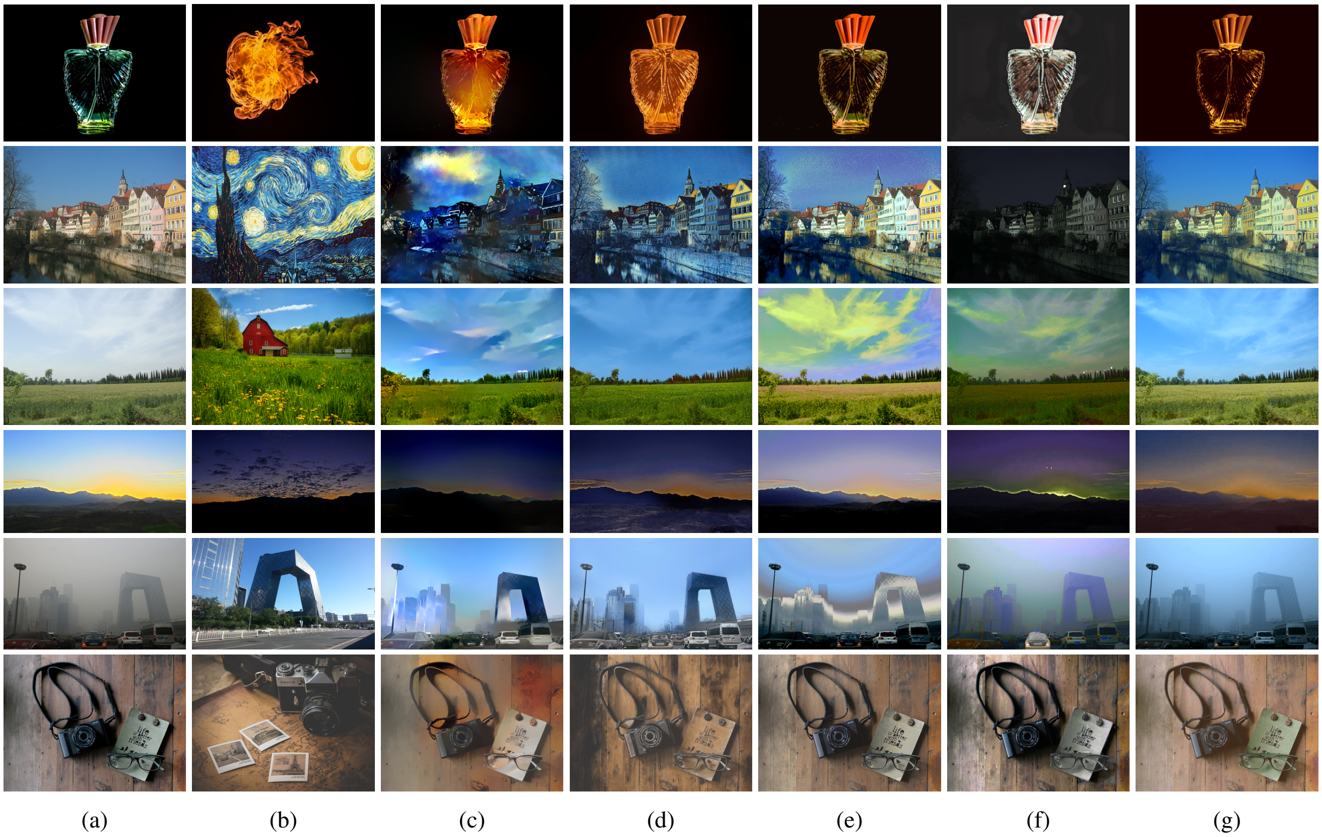


Figure 10: Comparisons with deep learning-based color transfer methods. (a) Original image; (b) reference image; (c) Luan et al. [26]; (d) He et al. [17]; (e) Yaosen et al. [44]; (f) Fang et al. [10] and (g) ours.

	P.R.	F.R.C.	C.B.
[31]	2.84±0.36	2.69±0.38	7.62%
[29]	3.01±0.52	3.14±0.56	12.86%
[13]	2.60±0.57	2.86±0.69	7.62%
[14]	2.79±0.39	3.23±0.33	12.86%
Ours	**3.86±0.28**	**4.01±0.20**	**59.52%**

(a)

	P.R.	F.R.C.	C.B.
[26]	2.81±0.73	3.51±0.59	12.38%
[17]	3.26±0.40	3.66±0.41	29.52%
[44]	2.83±0.81	2.80±0.71	4.76%
[10]	2.54±0.21	2.14±0.35	1.43%
Ours	**4.07±0.28**	**3.83±0.35**	**51.90%**

(b)

Table 2: User study results: (a) Comparison with traditional methods. (b) Comparison with deep learning-based methods. P.R.: perceptual realism, F.R.C.: faithfulness to reference Color, C.B.: choice of the best. The second and third columns display the average and standard deviation of the scores, and the fourth column indicates the percentage of times each method is chosen as the best by participants.

in this section. Our analysis reveals an average reconstruction error of 0.005 ± 0.005 and a maximum error of 0.017. These results indicate that the reconstruction error remains minimal when using the calculated coordinates for recoloring. Consequently, while negative values may occur in our coordinates, they do not adversely affect the visual quality of the results.

Time consumption. Our color transfer algorithm does not require the training process associated with deep learning methods. To perform color transfer, we first pre-extract the image's color palette, with the extraction of the simplified convex hull being the most time-intensive step. This duration varies depending on the number of initial convex hull vertices, which correlates with the image's color distribution. For the examples in this section, extracting the final palette of an image takes approximately 3 to 28 seconds. Once the palettes are obtained, our subsequent color transfer operations are comparable in time to traditional methods. For instance, in the second example shown in Fig. 9, the processing times for our method and those of [31, 29, 13, 14] are 1.62, 4.68, 0.02, 3.31, and 7.49 seconds, respectively. While our method requires additional time for palette extraction, it consistently delivers higher quality color transfer results.

6.3. Results of grayscale image colorization

In Fig. 11, we compare our grayscale image colorization results with five traditional methods, including the pioneering global statistics-based method [41], the user interaction-based method [21], the superpixel-based method [16], and two pixel-level methods [6, 28]. Observing the results, traditional methods may introduce noticeable artifacts (Fig. 11(c&f)) and struggle to transfer colors well between towers (Fig. 11(c,d,f&g)) or clouds (Fig. 11(e)). In contrast, our method adeptly transfers foreground and background colors to grayscale images. Additionally, we compare our method with three recent deep learning-based approaches, including [45], [19], and [38], in Fig. 12. Once again, these methods struggle to effectively transfer colors, especially in cases where the original and reference images are semantically distinct, as seen in the last row of Fig. 12. It is worth noting that while our method successfully transfers foreground and background colors to grayscale images with clear foreground objects, as shown in this section, it may encounter challenges with images where the foreground is intentionally blurred. To address this limitation and enhance versatility across various images, integrating a more advanced segmentation and correspondence method into our approach is a potential avenue for future improvement.

6.4. Results of video color transfer

Due to the robustness of our method, we have no special requirements for the test videos. Achieving temporally consistent video editing involves maintaining consistent colors for the same object across different frames. Fig. 13 showcases our editing results on a video. With a changing background, the clothes of the portrait maintain the same color style as the corresponding reference image. We compare our temporally consistent video color transfer results with those from [5] and [44] in Fig. 14. Notably, [5], which uses foreground-background segmentation during color transfer, exhibits artifacts in the contour of the character, as shown in the zoomed-in regions in Fig. 14(c). On the other hand, the method in [44] requires semantic correspondence between frames and the reference image; otherwise, it may result in unnatural artifacts in the background, as seen in the zoomed-in regions in Fig. 14(d). In contrast, our method achieves consistency in the color style across all frames and is free of artifacts or unnatural colors.

We apply time-varying editing to videos to achieve the effect of the same scene evolving over time and showcase examples in Fig. 15. Our goal is to transition the scene in the video from spring to winter, darkening the trees in the summer and reddening them in the fall. The results of [9] were generated by manually adjusting the color palette colors to closely match the colors of the video frames with those of the corresponding reference image, where the editing may not be as straightforward and requires a certain level of color perception from the user. Our result successfully achieves the desired effect by simply specifying the reference images, and the color from the reference image smoothly propagates to adjacent frames, resulting in a gradual color transition without abrupt changes. In addition, both [9] and we need to pre-process the video to generate the video color palette before editing the video. The computation of our video color palette is simple and efficient, hence is less time-consuming.

7. Conclusion and discussion

In this paper, we present an innovative color palette designed with a heightened level of representativeness, effectively conveying pixel distribution across the color space. This advanced palette seamlessly extends its applicability to videos. Additionally, we introduce a tailored algorithm for transferring colors from a reference image to the original, providing operational versatility. Our algorithm can function in a fully automatic mode or offer user interactivity at varying levels, accommodating coarse-to-fine editing. Furthermore, our approach easily extends to grayscale image colorization, temporal consistency and time-varying video color editing. Experimental results demonstrate the effectiveness of our method in achieving faithful color transfer from reference to input.

Our current method has certain limitations. Firstly, the palette extraction process is relatively slow and we plan to speed it up in the future. Secondly, utilizing a color palette for automatic color transfer may not effectively account for variations in image brightness or contrast. An alternative solution involves exploring different color spaces, such as Lab space, which reduces channel correlation and effectively represents the lightness channel. Handling the lightness channel separately during color transfer allows for improved control over the resulting image's brightness and contrast. Furthermore, in the results of our color transfer, the recoloring of certain examples exhibits good semantic correspondence. However, our automatic color transfer algorithm, primarily centered on global color information, does not have the capacity to ensure a match in semantic details between two images. While user interaction can alleviate this limitation to some extent, achieving complete semantic correspondence remains a challenge for the current algorithm. Future efforts may focus on enhancing semantic matching by integrating image texture information into our approach.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Nos. 62272402, 62372389), the Natural Science Foundation of Fujian

Figure 11: Comparison with traditional grayscale image colorization methods. (a) Original image; (b) reference image; (c) Welsh et al. [41]; (d) Irony et al. [21]; (e) Gupta et al. [16]; (f) Bugeau et al. [6]; (g) Pierre et al. [28]; and (k) ours.

Figure 12: Comparison with grayscale image colorization methods based on deep learning. (a) Original image; (b) reference image; (c) Yin et al. [45]; (d) Huang et al. [19]; (e) Wang et al. [38]; and (f) ours.

Province (Nos. 2024J01513243, 2022J01001), and the Fundamental Research Funds for the Central Universities (No. 20720220037).

References

[1] M. Afifi, M. A. Brubaker, and M. S. Brown. Histogan: Controlling colors of gan-generated and real images via color histograms. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7937–7946, 2021. 3

[2] Y. Aksoy, T. O. Aydin, A. Smolić, and M. Pollefeys. Unmixing-based soft color segmentation for image manipulation. *ACM Trans. Graph.*, 36(2), mar 2017. 2

[3] X. An and F. Pellacini. User-controllable color transfer. *Computer Graphics Forum*, 29(2):263–271, 2010. 1, 3

[4] B. Arbelot, R. Vergne, T. Hurtut, and J. Thollot. Local texture-based color transfer and colorization. *Computers & Graphics*, 62:15–27, 2017. 1, 3

[5] N. Bonneel, K. Sunkavalli, S. Paris, and H. Pfister. Example-based video color grading. *ACM Trans. Graph.*, 32(4), jul 2013. 3, 6, 11, 13

[6] A. Bugeau, V.-T. Ta, and N. Papadakis. Variational exemplar-based image colorization. *IEEE Transactions on Image Processing*, 23(1):298–307, 2014. 11, 12

[7] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein. Palette-based photo recoloring. *ACM Trans. Graph.*, 34(4), jul 2015. 1, 2, 3

[8] C.-K. T. Chao, J. Klein, J. Tan, J. Echevarria, and Y. Gingold. LoCoPalettes: Local control for palette-based image editing. *Computer Graphics Forum (CGF)*, 42(4):e14892, June 2023. 2

[9] Z.-J. Du, K.-X. Lei, K. Xu, J. Tan, and Y. Gingold. Video recoloring via spatial-temporal geometric palettes. *ACM Trans. Graph.*, 40(4), jul 2021. 2, 6, 11, 14

[10] Y. Fang, P. Yuan, C. Lv, C. Peng, J. Yan, and W. Lin. Saliency guided deep neural network for color transfer with light optimization. *IEEE Transactions on Image Processing*, 33:2880–2894, 2024. 3, 7, 8, 9, 10

[11] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol. Regularized discrete optimal transport. In A. Kuijper, K. Bredies, T. Pock, and H. Bischof, editors, *Scale Space and Variational Methods in Computer Vision*, pages 428–439, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 3, 7, 8

[12] D. Freedman and P. Kisilev. Object-to-object color transfer: Optimal flows and smsp transformations. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 287–294, 2010. 2

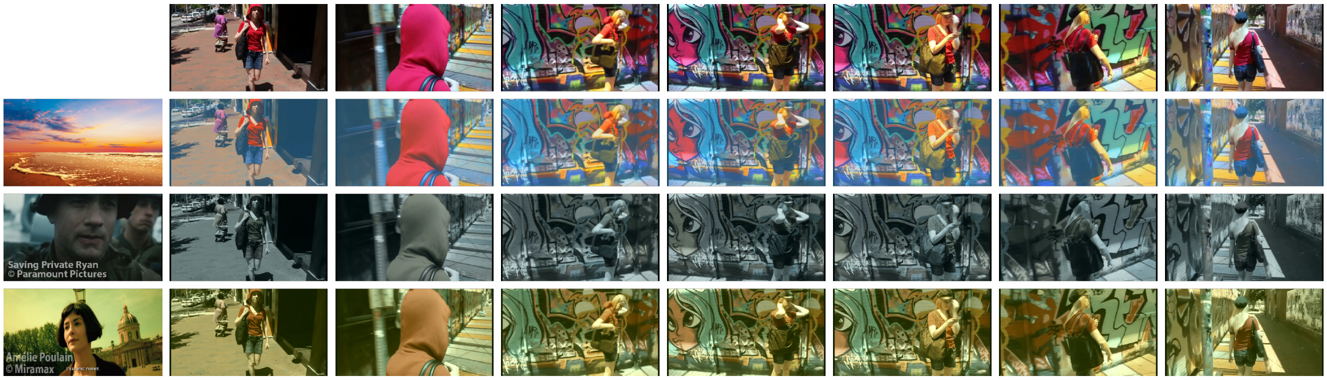


Figure 13: Our temporally consistent color transfer results. The first row is the video frames selected from the original video, and the other rows are the frames of the video after color transfer to the original video with reference image.

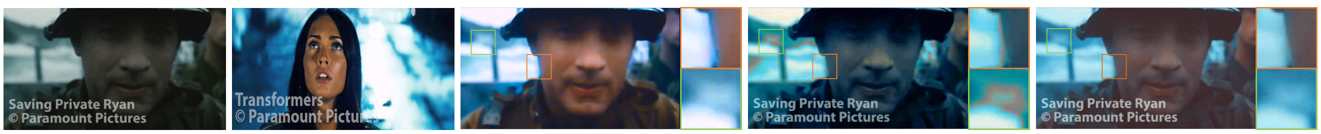


Figure 14: Comparison with Bonneel et al. [5] and Yaosen et al. [44]. Shown here are frames taken from the videos. (a) Original video frame; (b) reference image; (c) Bonneel et al. [5]; (d) Yaosen et al. [44]; and (e) ours.

[13] R. Giraud, V.-T. Ta, and N. Papadakis. Superpixel-based color transfer. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 700–704, 2017. 1, 3, 7, 9, 10

[14] M. Grogan and R. Dahyot. L2 divergence for robust colour transfer. *Comput. Vis. Image Underst.*, 181:39–49, April 2019. 1, 3, 7, 9, 10

[15] C. Gu, X. Lu, and C. Zhang. Example-based color transfer with gaussian mixture modeling. *Pattern Recognition*, 129:108716, 2022. 3

[16] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong. Image colorization using similar images. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, page 369–378, New York, NY, USA, 2012. Association for Computing Machinery. 11, 12

[17] M. He, J. Liao, D. Chen, L. Yuan, and P. V. Sander. Progressive color transfer with dense semantic correspondences. *ACM Trans. Graph.*, 38(2), apr 2019. 1, 3, 7, 8, 9, 10

[18] H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch. Style-aware robust color transfer. In *Proceedings of the Workshop on Computational Aesthetics*, CAE '15, page 67–77, Goslar, DEU, 2015. Eurographics Association. 3

[19] Z. Huang, N. Zhao, and J. Liao. Unicolor: A unified framework for multi-modal colorization with transformer. *ACM Trans. Graph.*, 41(6):1–16, 2022. 11, 12

[20] IBM. *IBM ILOG CPLEX Optimization Studio*. IBM, 2021. Version 12.10.0.0. Available from https://www.ibm.com/support/knowledgecenter/SSSA5P_12.10.0. 5

[21] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. EGSR '05, page 201–210, Goslar, DEU, 2005. Eurographics Association. 11, 12

[22] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.*, 33(4), jul 2014. 1

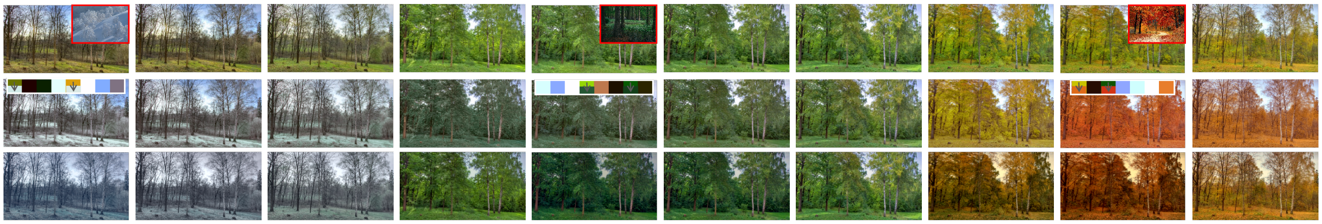
[23] J. Lee, H. Son, G.-H. Lee, J. Lee, S. Cho, and S. Lee. Deep color transfer using histogram analogy. *The Visual Computer*, 36:2129 – 2143, 2020. 3

[24] J. Li, J. Zhang, and D. Tao. Deep automatic natural image matting. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 800–806. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track. 6

[25] S. Liu and Y. Zhang. Temporal-consistency-aware video color transfer. In *Computer Graphics International Conference*, pages 464–476. Springer, 2021. 3

[26] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6997–7005, 2017. 1, 3, 7, 8, 9, 10

[27] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, page 169–178, New York, NY, USA, 2000. Association for Computing Machinery. 3



Frame 1 Frame 5 Frame 9 Frame 32 Frame 36 Frame 40 Frame 44 Frame 57 Frame 61 Frame 65

Figure 15: A comparison of the time-varing recolored result with Du et al. [9]. The first row is the original video frames, in which the first, 36-th and 61-th frames have been color transferred, and the reference image is shown in the upper right corner of the image. The second row is the results of Du et al. [9], which is produced by manually modifying the color palettes. The third row displays our results of color transfer for the corresponding frames.

[28] F. Pierre, J.-F. Aujol, A. Bugeau, N. Papadakis, and V.-T. Ta. Luminance-chrominance model for image colorization. *SIAM Journal on Imaging Sciences*, 8(1):536–563, 2015. 11, 12

[29] F. Pitie, A. C. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to colour transfer. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, page 1434–1439, USA, 2005. IEEE Computer Society. 1, 2, 7, 8, 9, 10

[30] J. Rabin and N. Papadakis. Non-convex relaxation of optimal transport for color transfer between images. In F. Nielsen and F. Barbaresco, editors, *Geometric Science of Information*, pages 87–95, Cham, 2015. Springer International Publishing. 3, 7, 8

[31] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001. 2, 7, 9, 10

[32] K. Siang Tan and N. A. Mat Isa. Color image segmentation using histogram thresholding – fuzzy c-means hybrid approach. *Pattern Recognition*, 44(1):1–15, 2011. 2

[33] N. Singla. Motion detection based on frame difference method. *International Journal of Information & Computation Technology*, 4(15):1559–1565, 2014. 6

[34] M. Solli and R. Lenz. Color semantics for image indexing. In *5th European Conference on Colour in Graphics, Imaging, and Vision and 12th International Symposium on Multispectral Colour Science*, pages 353–358. The Society for Imaging Science and Technology, 2010. 1, 2

[35] J. Tan, J. Echevarria, and Y. Gingold. Efficient palette-based decomposition and recoloring of images via rgbxy-space geometry. *ACM Trans. Graph.*, 37(6), Dec. 2018. 2, 3

[36] J. Tan, J.-M. Lien, and Y. Gingold. Decomposing images into layers via rgb-space geometry. *ACM Trans. Graph.*, 36(1), nov 2016. 2

[37] S. Waldron. Affine generalised barycentric coordinates. *Jaen Journal on Approximation*, 3(2):209–226, 2011. 4

[38] H. Wang, D. Zhai, X. Liu, J. Jiang, and W. Gao. Unsupervised deep exemplar colorization via pyramid dual non-local attention. *IEEE Transactions on Image Processing*, 32:4114–4127, 2023. 11, 12

[39] Y. Wang, Y. Liu, and K. Xu. An improved geometric approach for palette-based image decomposition and recoloring. *Computer Graphics Forum*, 38(7):11–22, 2019. 2

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 8

[41] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. *ACM Trans. Graph.*, 21(3):277–280, jul 2002. 11, 12

[42] S. Yan, S. Xu, W. Yang, and S. Y. Zhang. Image recoloring based on fast and flexible palette extraction. *Multimedia Tools and Applications*, pages 1–18, 2023. 2

[43] C.-H. Yao, C.-Y. Chang, and S.-Y. Chien. Example-based video color transfer. *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2016. 3

[44] C. Yaosen, H. Yang, Y. Yang, Y. Liu, W. Wang, X. Wen, and C. Xie. Nlut: Neural-based 3d lookup tables for video photorealistic style transfer. *ArXiv*, abs/2303.09170, 2023. 3, 7, 8, 9, 10, 11, 13

[45] W. Yin, P. Lu, Z. Zhao, and X. Peng. Yes, "attention is all you need", for exemplar based colorization. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 2243–2251, New York, NY, USA, 2021. Association for Computing Machinery. 11, 12

[46] I. Zabaleta and M. Bertalmío. Photorealistic style transfer for video. *Signal Processing: Image Communication*, 95:116240, 2021. 1, 3

[47] Q. Zhang, C. Xiao, H. Sun, and F. Tang. Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing*, 26(4):1952–1964, 2017. 2, 6

[48] Q. Zheng, M. Lu, S. Wu, R. Hu, J. Lanir, and H. Huang. Image-guided color mapping for categorical data visualization. *Computational Visual Media*, 8:613 – 629, 2022. 2