Image-Driven Harmonious Color Palette Generation for Diverse Information Visualization

Shuqi Liu, Mingtian Tao, Yifei Huang, Changbo Wang, and Chenhui Li

Abstract—Color has been widely used to encode data in all types of visualizations. Effective color palettes contain discriminable and harmonious colors, which allow information from visualizations to be accurately and aesthetically conveyed. However, predefined color palettes not only lack the flexibility of custom color palette generation but also ignore the context in which the visualizations are used. Designing an effective color palette is a time-consuming and challenging process for users, even experts. In this work, we propose the generation of an image-based visualization color palette to exploit the human perception of visually appealing images while considering visualization cognition. By analyzing color palette constraints, including harmony, discrimination, and context, we propose an image-driven color generation method. We design a color clustering method in the saliency-hue plane based on visual importance detection and then select the palette based on the visualization color constraints. In addition, we design two color optimization and assignment strategies for visualizations of different data types. Evaluations through numeric indicators and user experiments demonstrate that the palettes predicted by our method are visually related to the original images and are aesthetically pleasing, supporting diverse visualization contexts and data types in practical applications.

Index Terms—Visualization design, Color palette, Color assignment, Information visualization, Visual perception

1 INTRODUCTION

HUMANS are used to acquire information by observing the colorful world around them. Taking advantage of observations through sight, visualizations utilize color to convey the specific information inherited from the source data to target audiences. The quality of color encoding affects the effectiveness of information transmission. For example, the inappropriate usage of color mapping may cause problems such as confusion, obscuration, and actively providing misleading information [1]. In this work, we provide a preliminary hypothesis, assuming that color encoding is conditioned on the data, the color palette, the task, and the context. Specifically, the color palette represents many colors, which are used to encode the data. The task indicates what one wishes to convey, such as the trend or the pattern [2] of the underlying data. The context is where the visualization is used, such as a poster [3] or a projected slide.

The color palette is the basis of encoding visual expressions with color. A great color palette helps the visualization deliver information accurately with an aesthetically pleasing visual effect. Color harmony has a wide range of meanings and is described differently by different authorities [4]. There are many qualitative rules for color harmony. Some artists have defined forms, schemes, and relations in color space to describe harmonic colors [5, 6]. However, color harmony is also widely influenced by individual preferences, and decisions regarding aesthetic and harmonious palettes are personally different [7, 8]. Burchett et al. [4] stated that colors viewed together that produce a pleasing affective response are said to be in harmony. Similarly, Cohen-Or et al. described harmonic colors as a set of colors that are aesthetically pleasing in terms of human visual perception [9]. While there are some predefined color palettes available for reference, they lack the flexibility to accommodate diverse aesthetic preferences and visualization themes. Designing a harmonious and pleasing color palette for visualization from scratch is quite complex and requires professional knowledge. Some designers draw on images to obtain color-matching inspiration and improve efficiency. Image aesthetic value has been applied to image retrieval, image restoration, art design, advertising, and other fields. On the one hand, images are the most popular visual media and are widely used in information dissemination. Therefore, everyone can easily obtain images that meet their aesthetic preferences from social networks or websites. On the other hand, compared with predefined palettes, most of images contain richer visual information and have their own personalities, which means each of them has the potential to inspire unique design ideas. More importantly, images may already exist as references or constraints for the visual style of a visualization in a certain design scene [10]. Some image-driven color palette generation methods [11, 12, 13] have many application prospects, and we believe that color coding for visualization could also benefit from this design pattern.

Several studies have been performed to extract color from images. Unsupervised methods mainly include clustering methods [15], convex hull enclosure methods [18], and partition statistic methods [19] that focus on the color value at the pixel level. However, these studies merely regarded

• *Shuqi Liu, Mingtian Tao, Yifei Huang, Changbo Wang and Chenhui Li are with the School of Computer Science and Technology, East China Normal University, Shanghai, China.* *This work was supported by the NSFC under Grant 62072183 and the Shanghai Committee of Science and Technology, China (Grant No. 22511104600).* *Chenhui Li is the corresponding author. E-mail: shuqiliu234, yifeihuang17@gmail.com, 51205901082@stu.ecnu.edu.cn, cbwang, chli@cs.ecnu.edu.cn.*

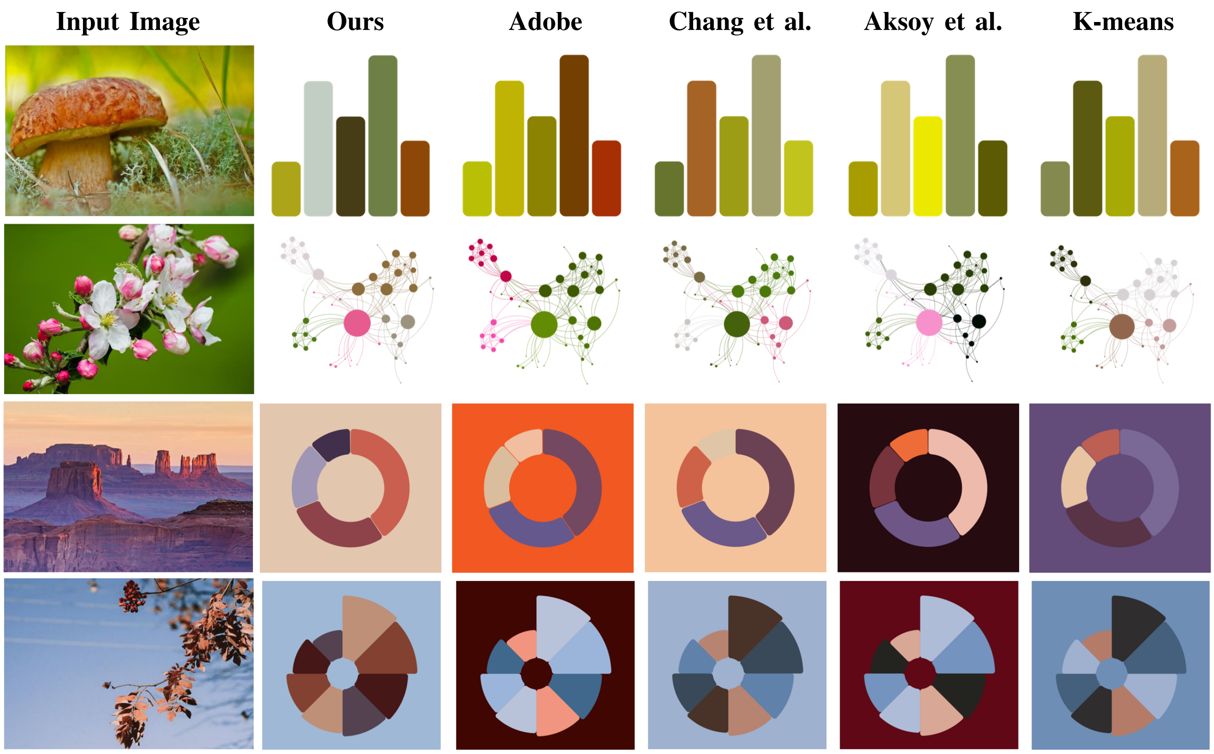


Fig. 1. Visualizations colored by resulting palettes extracted by our methods, Adobe [14], Chang et al. [15], Aksoy et al. [16], and K-means [17].

the palette as a feature of an image and do not consider the color requirements for visualization. For some state-of-the-art automatic color palette generation algorithms, [20, 21] measured visual discrimination metrics for palettes with distinguishing colors; [22] analyzed and modeled designer-crafted patterns for color ramps. These methods usually provide various adjustable parameters for users to customize palettes. However, it is not easy for their users to control the overall style to adapt to diverse visualization tasks and contexts. In addition, interaction-driven approaches [20, 21] are not suitable for incorporating automated, large-batch workflows.

To fill the gaps in the research described above, we propose an image-driven visualization palette generation method. It takes advantage of the human visual perception of images and can help users design a cognitive color palette for their target context. For data preprocessing, we first used a saliency detection model to distinguish between the subject and background in the image. We then used an FCN-based [23] superpixel segmentation network to reduce the dimensions while preserving the high-level features of the original image. In our framework, we designed a clustering method based on the visual saliency scores and hue values of colors in an image that can group colors at different fineness levels in regions with different visual importance. After obtaining the color candidate set, based on the requirement of color separability in visualizations, we modeled colors as normal distributions in the chroma-lightness plane, evaluating the color difference using the Bhattacharyya distance of the distribution and the hue value distance. For discrete data, our color selection process can produce a palette with a variable number of colors and a background color (optional) while ensuring that the colors are distinguishable. Then, to bring the results in line with human aesthetic perception, we optimized the palette based on a series of aesthetic principles. For continuous data, we generated continuous diverging palettes with a perceptually linear change rate. Finally, we recommended different color combinations and assignment strategies based on the context type of the visualization.

Our method can be applied to various practical visualization application scenarios. Compared to other image-based color extraction methods [14, 15, 16], our results contain distinguishable colors and are more similar to the original image. Some comparison instances are shown in Figure 1. In addition, we provide users with a new perspective on generating visualization palettes, allowing them to select vivid natural images instead of dull parameters to constrain the result. Our method can also be used in an automated visualization design workflow. We demonstrated the applicability and reliability of our method through experiments on actual visualization scenarios, the evaluation of quantitative indicators, and user studies. Our contribution is mainly in the following three aspects:

(1) We define the problem of color design for visualization in an image-driven manner for the first time and design a framework for solving this problem.
(2) We propose an image-driven visualization palette generation method based on human visual and aesthetic perception.
(3) We propose a context-aware color assignment strategy

that can automatically adapt to multitype and multistyle charts with different contexts and apply it to several real applications.

2 RELATED WORK

In this section, our survey of works related to color palette extraction, design, and color encoding choices is presented. Our discussion covers methods of extracting color palettes from media, as well as some important factors in visualization palette design and how different visualization tasks affect the color encoding choice.

2.1 Palette Extraction from Media

A color palette is composed of several colors. It can be utilized in the transmission of visual information and palette-based human interaction. Color palettes are usually chosen by designers or generated with the assistance of algorithms. We will focus on the research on extracting palettes from visual media, such as images and videos.

The aim of color palette extraction is to find several colors that represent the source visual media. Several methods have been proposed for different purposes. Utilizing an unsupervised method for the color transfer of images, Chang et al. [15] proposed computing $(K+1)$-means rather than k-means [17] to avoid choosing dark colors. However, the generated palette tends to cause a high level of inner similarity. Therefore, an attenuated item was introduced in [11] to reduce this problem. To achieve color transformation in the *a*-*b* plane of the CIELAB color space, Huang et al. [24] proposed the H-means method to generate an appropriate color palette. In contrast to extracting colors, [16] extracted the color distribution represented as Gaussian kernels generated from several seed pixels.

Overall, the recent popular color palette extraction unsupervised methods can be divided into 3 categories. The first is color clustering algorithms, including the work in [11, 25], which groups colors into several clusters based on the task-driven distance. The second is convex hull enclosure solutions, such as employing convex hull geometry to represent images [18]. The last are partition statistic approaches that rely on mask [16] and histogram thresholding [26]. The advantages of both color clustering and convex hull enclosure were fused in [12] to achieve color palette extraction from movies. To capture topological changes in a video's palette, [27] built the RGBT skew polytope based on the RGB convex hull of each frame. Weingerl et al. [13] built a LASSO regression model trained on human-extracted color themes to extract prominent colors. These methods are more in line with human theme color extraction and can be used to observe the colors of small but interesting parts. [28, 29] extracted a color set from images and optimized it using the evaluation models trained on human-made palette scoring datasets.

The color palette extraction methods mentioned above regard the palette more as an image feature rather than a guide for color design in visualizations. As a result, these methods have not utilized the requirements for visualization color and cannot achieve image-based visualization color encoding.

2.2 Color Encoding for Visualization

The goal of color design for visualization is to provide an appropriate color palette to ensure that visualizations are accurately and aesthetically created. To maximize the information transmission underlying the visualization work, Borland et al. [30] encouraged visualization designers to work closely with domain experts. In [31], static images of visualization were fed into the deep learning network to extract their color maps. Healey et al. [32] and Tufte et al. [33] provided practical guidelines according to the constraints of human visual perception for distinguishable colors. Maxwell et al. [34] pointed out that the spatial distribution of colors is also a condition for a discriminable palette. To make the calculated maximum scaled difference more in line with the human visual perception distance, Lu et al. [21] optimized algorithms that maximize color distance. However, these methods only pay attention to the distinction between colors and neglect the aesthetic quality of visualization.

As [35] mentioned, an enjoyable visualization task benefits from its visual aesthetics, and color design is a crucial part of this task. They suggested taking advantage of the aesthetic and attention-guiding cognition of professional designers and artists to create a knowledge-based color design system. Gramazio et al. presented Colorgorical [20] for creating color palettes based on user-defined discriminability and preferences. Using a linear regression model-based algorithm, they calculated the aesthetic preference score to select colors from the CIELAB space. Smart et al. [22] fit an input color to an aesthetically pleasing color ramp by modeling designer-crafted gradient color palettes to obtain a mimicked result. However, these methods mostly focused on making color palettes harmonious while rarely considering the visualization context. Ahmad et al. [36] mentioned that in addition to aesthetic qualities, colors in data visualizations can impact viewers' cognition, affect, and behavior through semantic association. Samsel et al. [37] extracted semantically appropriate color palettes from natural imageries for scientific visualization and proved that they can be used to improve analysis and communication. These works illustrated that thematically and semantically relevant colors can produce better expressiveness in visualizations. Our image-based approach can automatically generate aesthetically pleasing context-aware palettes, enabling visualization work that reflects the theme and visual style of the corresponding image.

In addition, the data type is also an important factor in visualization color coding. In general, there are the following four data types: nominal, ordinal, interval, and ratio. Nominal data can represent categorical variables, and [32] selected an effective color palette for nominal data by controlling the following three metrics: color distance, linear separability, and category. The concept of class visibility was proposed to reduce the contrast effect for better distinguishability

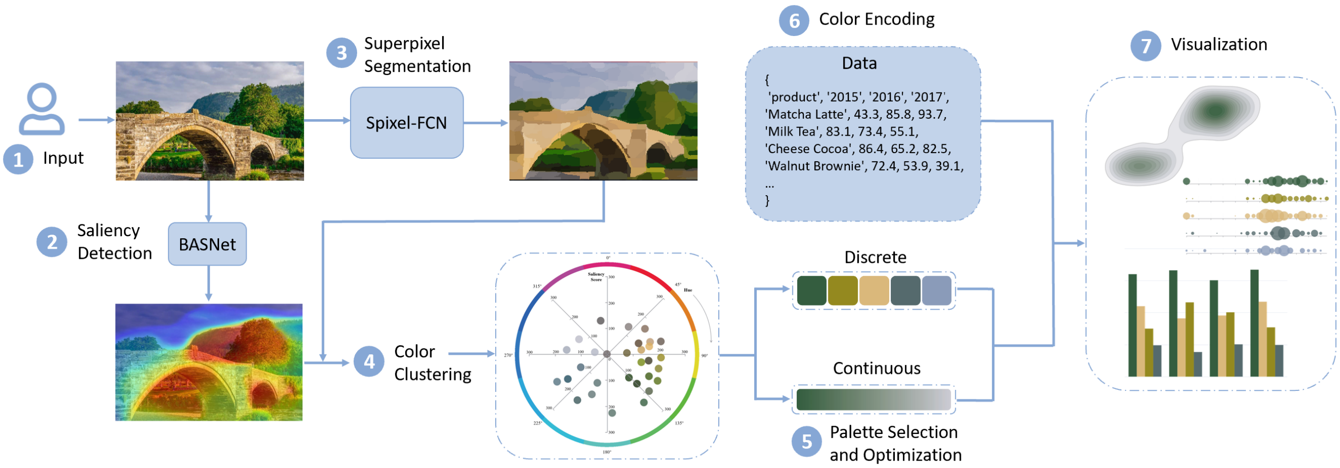


Fig. 2. The pipeline of our method. 1) Users choose an image as the input for color scheme reference. 2) The saliency detection for the input image is performed and the visual importance levels of different regions are distinguished. 3) We perform superpixel segmentation as the preprocess to overcome the negative influence of image quality and facilitate subsequent calculations. 4) The average saliency score of each segmentation area is calculated and then colors in the saliency-hue plane are grouped to obtain a candidate set. 5) The palette is selected and optimized based on a series of color aesthetics principles. 6) The colors for visualizations are encoded according to different data types. 7) Visualization of colored results.

performance [38]. Compared to nominal data, the orders of ordinal data matter. Therefore, monotonic data can be represented by the degree of lightness [1, 39]. Nominal and ordinal data are discrete, while interval and ratio data are continuous. One basic technique for encoding continuous scalar data is to apply monotonic luminance profiles [40].

However, visualization color encoding still depends on some other aspects. For example, whether one wishes to express metric information or form information [41] should affect color encoding. Several techniques were introduced in [42] to compensate for target tasks, including lookup, comparison, and relation seeking. The perceptually based hue-chroma-luminance (HCL) color space was proposed in [43] to encode all types of data. Later, HCL was employed in [44] to generate tree-specific color palettes for improving node-link diagrams and revealing tree structures. Liu et al. [45] used colors of different chroma to represent the activation situation of neurons in a CNN. Directly optimizing the control points of the color palette [2] can be employed to enhance the boundary characteristics of the data. However, these studies rarely applied the color assignment from a predefined palette and cannot be widely applied to visualization scenarios without further improvements.

3 Overview

As the carrier of information, visualization provides a concise and distinct way of displaying the features of data. Well-designed colors are aesthetically pleasing and increase the cognition and perception effectiveness of the visualization. A complete color image can provide much more information and inspiration than a palette containing limited colors. For those with little experience in palette design, beautiful images with harmonious tones are good references. Professional designers tend to draw inspiration from complete images. In addition, the tonal style of a visualization instance can match a specific theme to deepen viewers' impressions by extracting colors from representative images. The tonal style is also suitable for situations where a chart needs to be embedded in an image or when charts and images are presented together.

To automatically generate high-quality color palettes from images, we designed a saliency-hue-based color cluster method to extract colors and used a palette harmony optimization algorithm in the lightness-chroma plane to generate the final results. The method architecture is shown in Figure 2. Our method represents colors in CIELCh color space with lightness (L), chroma (C), and hue (h).

First, we used a deep neural network model to detect and simulate the important visual features of I and generate a visual importance map I_{sa} that contains saliency scores for all colors. The subsequent color assignments generated were closer to people's perception habits. Second, we used a superpixel segmentation network to divide pixel-level images I into district-level results I_{sp} for a more general image representation. Third, we took the saliency score as the radius and mapped the hue value of color to $[0°, 360°)$ as the angle to define a circular distribution, which we called the saliency-hue circular plane. Then, we calculated the average saliency score of each district-level color based on I_{sa} and I_{sp}, distributing them in the saliency-hue plane for clustering to obtain a palette candidate set D. Then, for discrete data, we designed a selection process that prioritizes colors with higher saliency values to select the initial palette Pi for secreting data from the candidates set D while ensuring that the colors are not similar to one another. For ordinal data and continuous data, the palette should have good perceptual resolution and be perceptually uniform throughout. We selected two colors from the candidate set D, considering the saliency and color distribution in image I. We followed the InterpolateColor algorithm in [46] and obtained a continuous diverging palette with a perceptually linear change rate.

4 METHODS

4.1 Saliency-based Multilevel Perception

When extracting the theme color palette from an image, the overall information of the image should be considered, which is the simplest way to pay equal attention to each part of the image in most cases. However, it should also be realized that people will focus on the main object of the image, which means that the color details of partial areas with high visual importance should receive more attention.

The choice of the most important region by the human eye depends not only on some low-level features of the image, such as contrast, shape, size, and color but also on high-level semantic-related features of the image. Therefore, we introduced a saliency detection network based on deep learning to adapt to the human perception mechanism of the region with high visual importance.

Traditional methods detect important objects based on handcrafted features and calculate the predefined measure pixel by pixel [47]. After the invention of Deep CNNs [48], researchers started to use CNNs to extract features from images. FCN-based methods [49] are widely used in many tasks because they can adaptively capture high-level semantic information. However, the lack of an upsampling process gives rise to missing low-level features, especially the object boundary.

We adopted the state-of-the-art salient detection network BASNet [50] shown in 3(a) to construct the visual importance map. This network uses an encoder-decoder network to obtain rough results and then uses a residual refinement module to fix the edges of the segmentation results. This architecture ensures both the region accuracy and the boundary quality. We simplified the original hybrid loss function, keeping only the term corresponding to the pixel-level feature as follows:

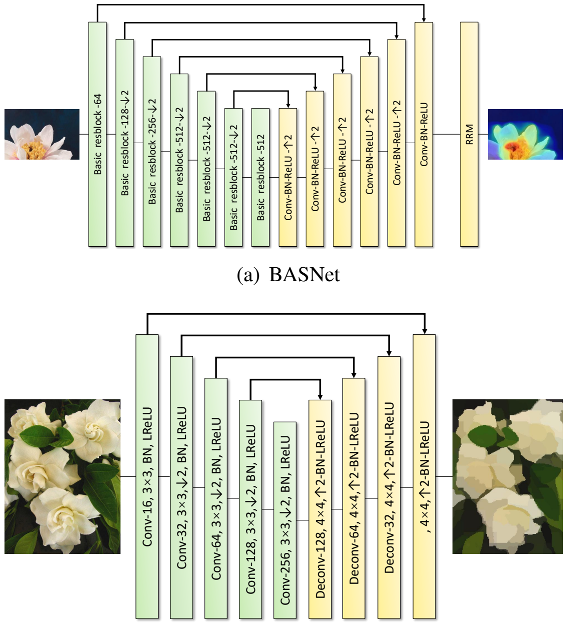
$$\ell_{saliency} = \ell_{bce} + \ell_{sim}, \tag{1}$$

where ℓ_{bce} and ℓ_{sim} represent BCE loss [51] and SSIM loss [52], respectively.

BCE loss [51] was used to measure the binary cross entropy between the ground truth label and the predicted visual importance map in the training process. The pixel index of the original ground image was $p = \{1, \ldots, N\}$, and the value of pixel p in the ground truth and predicted saliency maps were denoted as $G_p \in [0,1]$ and $S_p \in [0,1]$, respectively. Then, ℓ_{bce} was formulated as follows:

$$\ell_{bce}(S,G) = -\sum_{p}(G_p \cdot \log S_p + (1-G_p)\cdot \log(1-S_p)). \tag{2}$$

SSIM loss [52] measures the structural similarity between images. The visual importance map should keep the structural



(a) BASNet

(b) SpixelFCN

Fig. 3. Structures of the salient detection network BASNet and superpixel segmentation network SpixelFCN.

information from the ground truth, so our training integrated SSIM loss. We cropped corresponding patches from the gray ground truth G and predicted the visual importance map S. The gray values for G and P were $g = \{g_1, \ldots, g_N\}$ and $s = \{s_1, \ldots, s_N\}$, respectively. The SSIM loss was defined as follows:

$$\ell_{sim}(S,G) = 1 - \frac{(2\mu_g\mu_s + C_1)(2\sigma_{gs} + C_2)}{(\mu_g^2 + \mu_s^2 + C_1)(\sigma_g^2 + \sigma_s^2 + C_2)}, \tag{3}$$

where μ_g and μ_s denote the mean, σ_g and σ_s denote the standard deviation, and σ_{gs} denotes the covariance between g and s. $C_1 = 0.01^2$ and $C_2 = 0.03^2$ are scalar constants.

The dataset used for training was the GDI [53], in which the high visual importance regions are annotated and different saliencies of designed graphics are labeled. The model trained by GDI is able to consider all parts of the images and has a gentle saliency gradient change, which can finely divide the image into multiple saliency levels. We trained the network using the Adam optimizer for 25,000 iterations with a learning rate of 0.001 and a batch size of 8.

After obtaining the visual importance map of the input image, we extracted the representative color of the image background using the histogram method. Our method is based on the following two assumptions: (1) the background part of an image has a low-saliency score and (2) for the background part of an image, the inside color with the most pixels can represent the background best. We created a $16 \times 16 \times 16$ histogram for the L, C, h channels of color and computed the mean color to represent each bin. Then, pixels with saliency values less than 0.2 were assigned to bins, and the representative color of the bin with the most pixels was chosen as the background color bcg. It might seem somewhat arbitrary to set the threshold at 0.2, but it is an empirical value and is able to obtain good results in practice.

4.2 Superpixel Segmentation

When reducing the color space, it is more important to choose colors that represent some structural features of the original image than to accurately distinguish the specific value of each pixel. Due to the possible noise or unnecessary texture, such as jitter caused by image compression, a more sensible approach is to generalize the color of a structural region to resist negative influences. This process can also reduce the complexity of the algorithm and improve the efficiency of the framework.

Superpixel algorithms transform a pixel-level image into a district-level image. Graph-based algorithms [54] segment superpixels with graph-partitioning approaches. In clustering-based algorithms [55], neighboring pixels are clustered into unified classifications according to low-level features. However, in deep-learning-based methods [56], the semantic feature benefits can be perceived by the characteristic of the deep neural network, which obtains higher efficiency and retains more structural details.

SpixelFCN [57] uses a simple encoder-decoder structure, such as 3(b) based on FCN. The segmentation results using SpixelFCN achieved state-of-the-art performance on the BSDS500 benchmark dataset [58] and attained a very fast speed of 50 fps on the NYUv2 benchmark dataset [59]. To reduce unnecessary calculations, SpixelFCN focuses on local information when generating superpixels, which means that pixels only focus on their nearby area. Finally, a soft pixel-superpixel association map is generated, mapping every pixel to the superpixel domain with the highest probability. For pixel p, $f(p)$ denotes the properties that should be preserved. Let p' and $f(p)'$ be the location and property of the reconstructed superpixel, respectively. The loss function can be formulated as two terms, as follows:

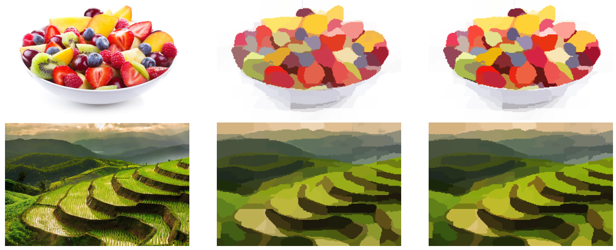
$$\ell_{spixel} = \sum_{p} E(f(p), f'(p)) + \frac{w}{S}\|p - p'\|_2. \tag{4}$$

The loss function enforces semantic property similarity in each group and the spatial compactness between groups individually. Here, E denotes the cross-entropy as a semantic distance measure, S is the superpixel interval, and w represents a weight for balance.

A representative color needs to be chosen for each region in the superpixel segmentation result. We found that simply calculating the average value of each pixel reduced the global contrast, brightness, and saturation. Let S_k be the k-th superpixel district and $c_k = \{c^k_{i,j} | (i,j) \in S_k\}$ be the color value of each pixel in S_k. Similar to [60], we used an adaptive coloring policy calculated by the average and median of c_k:

$$c_{i,j} = \left(\alpha_1 * \overline{c_k} + \alpha_2 * \widetilde{c_k}\right)^{\lambda}, \tag{5}$$

$$(\alpha_1, \alpha_2) = \begin{cases} (0,1) & \sigma(c_k) < \gamma_1 \\ (0.5, 0.5) & \gamma_1 < \sigma(c_k) < \gamma_2 \\ (1,0) & \gamma_2 < \sigma(c_k), \end{cases} \tag{6}$$



(a) Original image (b) Average color (c) Adaptive color

Fig. 4. Comparison of average color and adaptive color superpixel segmentation results.

where $\sigma()$ refers to the variance, $\gamma_1 = 20$, $\gamma_2 = 40$, and $\lambda = 1.2$. Figure 4 shows the comparison of average color and our adaptive color results in superpixel segmentation.

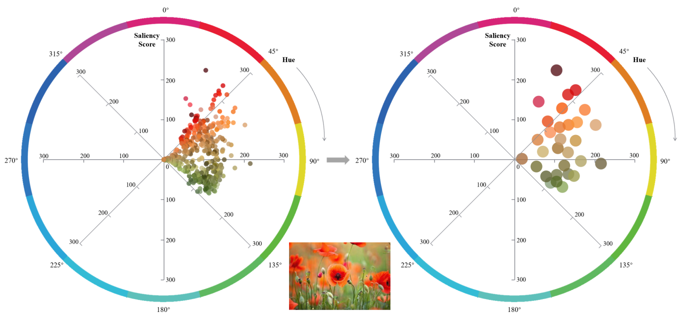
4.3 Color Clustering

We designed a clustering method that gathers the colors with similar hue and saliency scores from the district-level color set grouped by the superpixel segmentation result to obtain the palette candidate set.

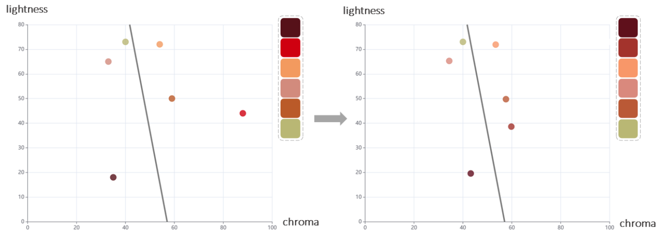
Let $RC = \{rc_0, rc_1, \ldots, rc_M\}$ be the representative color for each district. We calculated the average saliency score $s\bar{a}_i$ for the i-th district based on RC and the result of the salient detection network. Then, $s\bar{a}_i$ and the hue value $hue_i \in [0°, 360°)$ of the color in CIELCh space were taken as the radius and angle, respectively, to create a circular distribution, which we called the saliency-hue plane. When distributing all district-level colors of an image in the plane, the colors with high visual importance were closer to the outer edge.

Then, we used the position of the colors in the plane to cluster colors, generating the candidate set of the initial palette. Assuming the final palette has n colors, we set $n \times 5$ as the number of cluster centers. The distance between each point and its cluster centroid was specified as the L2 distance. We use k-means to group the colors with similar hue values and average saliency scores and then add the color with the highest saliency score of each cluster center to the candidate set. An example of the color distribution and clustering results in the saliency-hue plane is shown in Figure 5.

As it benefited from the geometric properties of the saliency-hue plane, the division of hue in high-salience areas (i.e., near the outer edge of the circle) was finer than that in low-salience areas (i.e., around the circle center), such as the background. This finding is in line with the visual perception habits of people when observing images. As shown in Figure 6, we extracted 30 colors from each image using our method and k-means based on the L2 distance in the RGB color space. It can be observed that our method obtained finer colors from the foreground of images. In addition, compared with the histogram statistics and distance-based color clustering methods, our method could distinguish color blocks with different saliency but similar hue in an image, which helps eliminate the interference of different subjects with similar colors on the output saliency. We further illustrated these advantages in detail with an example in the Appendix.



(a) Color Cluster in Lightness-Chroma Plane



(b) Palette Optimization

Fig. 5. The color clustering and optimization process. The angle between the color point and the positive direction of the y-axis in (a) is the hue value, and its distance from the origin is the saliency score. The hue ring can be used to assist in the observation.

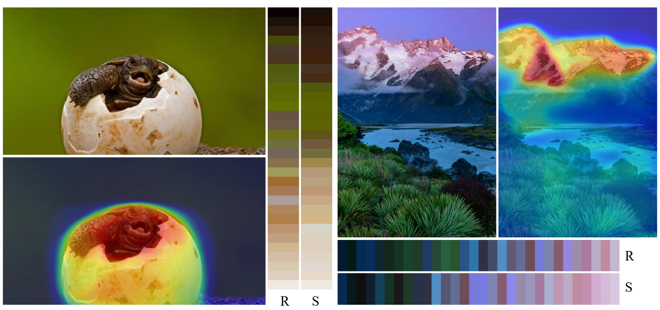


Fig. 6. Extracting 30 colors from images using k-means in the RGB color space (R) and saliency-hue plane (S).

4.4 Palette Selection and Optimization

In this section, we generated discrete and continuous palettes to accommodate different data types from the color candidate set.

4.4.1 *Discrete Palette*

In the process of selecting discrete palettes, we prioritized colors with higher saliency scores and expected that they were far away from one another. Considering a color point ψ in CIELCh coordinates, we used $L \in [0,100]$, $c \in [0,100]$ and $h \in [0°,360°)$ to represent its lightness, chroma and hue, respectively. The color distance we defined includes the following two parts: one is BD, representing the difference in chroma and lightness dimensions calculated by the method in [61], and the other is the hue difference HD. To describe the hue uncertainty estimation of ψ, we modeled (L,c) as a bivariate normal distribution, as follows:

$$T \sim \mathcal{N}([c,L]^\top, \sigma_{cL}), \sigma_{cL} = \begin{pmatrix} \alpha_c^2 S_c^2 & 0 \\ 0 & \alpha_L^2 S_L^2 \end{pmatrix}, \tag{7}$$

where α_c and α_L are constants and

$$S_c = 1 + 0.045c, \tag{8}$$

$$S_L = 1 + \frac{0.015(L-50)^2}{\sqrt{20+(L-50)^2}}. \tag{9}$$

Given two colors C_i and C_j and their normal distributions $T_i \sim \mathcal{N}([c,L]_i^\top, \sigma_i)$ and $T_j \sim \mathcal{N}([c,L]_j^\top, \sigma_j)$, respectively, we instead used the Bhattacharyya distance BD in the color difference formula [62] to evaluate the proximity. Here, BD is defined as follows:

$$BD = \frac{1}{8}(\Delta[c,L]^\top)^\top \sigma^{-1}(\Delta[c,L]^\top) + \frac{1}{2}\ln\left(\frac{\det \sigma^{-1}}{\sqrt{\det \sigma_i \det \sigma_j}}\right), \tag{10}$$

where $\Delta[c,L]^\top = [c,L]_i^\top - [c,L]_j^\top$ and $\sigma^{-1} = \frac{\sigma_i+\sigma_j}{2}$. It is considered in [61] that C_i and C_j are not ambiguous if $BD(T_i,T_j) \geq 3$.

Measuring the color distance with BD can help determine whether neutral colors with small chroma are distinguishable. We regarded $BD(T_i,T_j) \geq 3$ as one of the filter conditions to avoid ambiguous colors from appearing. Algorithm 1 describes the specific selection steps.

Algorithm 1: Selecting the palette from the candidate set

Data: $\Psi = \{[(L_i,c_i,h_i),s\bar{a}_i] \mid i = 0,\ldots,n\times 5\}$: a list of n×5 colors in CIELCh coordinates with their saliency scores $s\bar{a}_i$, sorted by $s\bar{a}_i$ in descending order.
Result: Color palette P with n colors selected from Ψ sorted by saliency scores in descending order.

```
initialization: P = Ψ[0];
for i ← 1 to n × 5 do
    if |h_i − h_j| > 5°, BD(T_i, T_j) < 3 with
      P = {[(L_j, c_j, h_j), sā_j] | j = 0, ..., P.length} then
        Add Ψ[i] = [(L_i, c_i, h_i), sā_i] to P;
        if P.length is equal to n then
            break;
        end
    end
end
if P.length < n then
    Divide the sorted remaining candidate colors into
      n − P.length groups. For each group, choose the
      color that has the largest hue difference with all
      elements in P.
end
Sort P by the saliency scores in descending order;
return P
```

After generating the initial palette, we aimed to optimize this palette to make it more harmonious while maintaining its visual style. For palettes in the Munsell color space, Matsuda defined several harmonic templates [5] in the value and chroma plane. Inspired by these templates, [61] proposed that colors in a harmonic palette must be in the vicinity of a line in the chroma-lightness plane. In this flexible optimization and evaluation method, there is no need to change the hue value of the color, which is an additional advantage.

We obtained a point set $\Theta = \{(c_i, L_i) \mid i = 0, \ldots, n\}$ of a palette with n colors on the lc plane and used it to fit to a line in polar coordinates $\hat{l} = (\hat{r}, \hat{\varphi})$; $\hat{r}$ and $\hat{\varphi}$ are parameters that need to be calculated using the following equations:

$$\hat{r} = \bar{c}\cos\hat{\varphi} + \bar{L}\sin\hat{\varphi}, \bar{c} = \frac{\sum \omega_i c_i}{\sum \omega_i}, \bar{L} = \frac{\sum \omega_i L_i}{\sum \omega_i}, \tag{11}$$

$$\hat{\varphi} = \frac{1}{2}\arctan\frac{-2\sum_i \omega_i(\bar{L} - L_i)(\bar{c} - c_i)}{\sum_i \omega_i[(\bar{L} - L_i)^2 - (\bar{c} - c_i)^2]}, \tag{12}$$

$$\omega_i = (\alpha_c S_c \alpha_L S_L)^- 2, \tag{13}$$

where $\bar{c}$ and $\bar{L}$ represent the mean values and ω_i is the weight of each measurement. α_c, S_c, α_L, and S_L are the same as defined in Equation 7. We calculated the L2 distance MD between l and each point in Θ as follows:

$$MD = |c_i \cos\hat{\varphi} + L_i \sin\hat{\varphi} - \hat{r}|. \tag{14}$$

If a color point is far from the line, we modified its chroma and lightness value to bring them closer together. To strike a balance between the harmony of the result and the degree of color change, we heuristically set the threshold of MD to 15. An example of this optimization is shown in 5(b). It is proven in our evaluation that the optimized color palettes are more harmonious and still have a strong relation to the theme of the original images.

4.4.2 Continuous Palette

Continuous visualization palettes can be divided into the following two types: sequential and diverging [63]. Most of the sequential types are monochromatic, showing a uniform gradient from a heavily saturated color to white. The diverging types usually have two major color components, and they transition from one color component to the other by passing through an unsaturated color (white or yellow) [46]. Compared to sequential color maps, diverging ones are more colorful; therefore, they can better represent the visual style of an image and are more aesthetically pleasing. Thus, we extracted a diverging continuous palette from the image.

Considering that diverging palette should include two major color components with good perceptual resolution and have a correlation with the picture, we selected two colors from the image from the aspects of saliency and color distribution while ensuring that the color distance is large enough. To obtain a continuous visualization palette with a perceptually uniform color gradient, we performed a calculation with the two colors following the InterpolateColor algorithm [46].

4.5 Color Assignment

Finally, we assigned the colors in the palette to visualization instances. For charts with nominal data, we used a discrete palette and assumed that the colors with larger areas have greater impacts on the overall color style. Thus, we assigned colors with higher saliency scores to larger areas. Compared to a random color assignment strategy, this strategy revealed that the theme of the visualization instance is closer to the original image theme. For ordinal data and continuous data, we recommend that users linearly map the colors in the continuous palette to the data since the palette has a perceptually linear change rate.

In addition, it is recommended that the background color *bcg* be utilized in different ways according to the specific chart context, as follows: (1) when the chart does not require a background, use *bcg* instead of the color with the lowest saliency score in the discrete palette; (2) when the chart needs a solid color as a background, use *bcg*; and (3) when the chart has to be embedded within the original image, ignore *bcg* to avoid unclear data in the image.

5 Applications

Image-driven methods can greatly simplify workflows and can be used in a wide range of applications. Using these methods, particular color palettes that are adapted to different contexts for diverse visualization tasks and design colors in infographics are generated. A series of related experiments were conducted with an Intel Core i7 CPU and an NVIDIA GeForce 2080 Ti GPU.

5.1 Image-based Palette Generation

Our image-based palette generation method is user-friendly and appropriate for new visualization designers. For novices, it may take a long time to design a palette without any reference. Although there are some predefined color palettes, they are usually generic and cannot satisfy different visualization themes or personal preferences. Most of the existing tools for automatically generating color palettes are not designed for visualization [14, 64], so their results cannot fit specific data or chart types. For the tools dedicated to visualization [20, 21], users need to set initial parameters that rely on their experience or manually modify colors during the generation process, which may cause them to lack inspiration when designing multiple palettes. Our method does not require users to have design experience; they only need to provide reference images and choose the data type. As a result, people can easily obtain images conforming to their aesthetic color preferences from the internet. Therefore, our method can alleviate problems caused by a lack of inspiration. In addition, color schemes need to be changed frequently due to different display scenes, contents, and even seasonal themes. It would be time-consuming and labor intensive to manually design colors for each scene. If the designer has ready-made product images, advertising, and environmental images for reference, our system can automatically generate palettes from images and thus provide both convenience and creative inspiration.

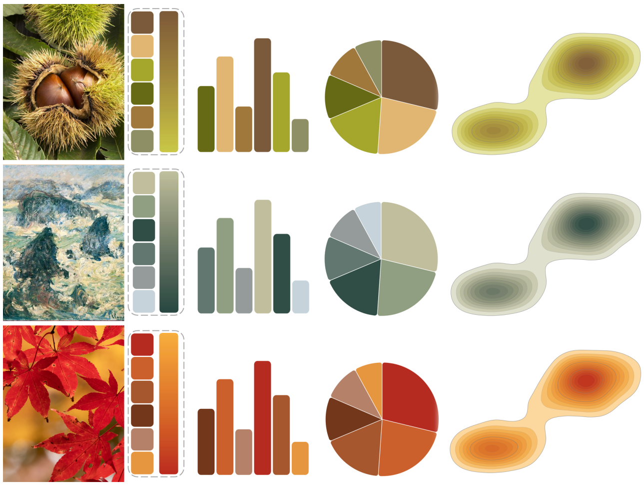


Fig. 7. Results of three visualization tasks using discrete and continuous datasets.

To demonstrate that our method is effective, we selected two types of datasets and formulated three visualization tasks. The first dataset contains discrete data with six categories. We used a **bar chart** and **pie chart** to show the numeric values for each category. Another continuous dataset contains 272 coordinate points, and their density is represented by a **heatmap**. Assuming that all of these visualization tasks are in a pure context with a white background, Figure 7 shows our results of the above tasks.

Our method works well for these tasks. The automated generation process provides vital ease of use, and the visual style of the palette is constrained by the original image. Benefiting from our clustering and selection strategy, the colors in the palette are not similar and satisfy a certain harmonic template in the chroma and lightness plane, which makes them aesthetically pleasing and harmonious. For discrete data, we assign colors according to their saliency scores in the original image to emphasize the visual center. For continuous data, our interpolated palette allows the visualization result to represent the data. Our image-driven method can be used by novices to quickly generate personalized visualization palettes and can also serve as a high-quality reference for experienced designers to help them obtain inspiration from images and improve their work efficiency.

5.2 Palette Generation for Diverse Tasks and Contexts

In addition to extracting palette references for visualization from images, our method can also be used flexibly in various situations. For example, when visualization charts are required in a commercial advertisement or a promotional website, they are often placed in a certain context and displayed with images. Choosing a color palette with strong relevance to the image for the chart helps the visualization appear more harmonious, and the image-based palette generation method works well for such problems. Specifically, we noticed that assigning colors can greatly increase the visual correlation between the charts and images if their background colors are similar, and we have proven this in the subsequent evaluation section. Compared with other image-based palette generation methods, our work can be used to distinguish the background color and calculate the saliency scores of each color in the result by using histogram statistics and a salient detection network. As a result, we can assign colors in a unique way that significantly improves the result.

Sometimes graphic designers embed charts as elements into the images on posters and in advertisements to display them as a whole, and our system also works for these cases. To avoid obstructing the area with high visual importance, the chart is generally placed in the background part of the image. When using image-based palettes for this situation, it is necessary to remove the background color to avoid poor data representation due to low contrast between the chart and image. Figure 8 shows the results of our system in the above cases. The results have a harmonious visual effect when displayed together with the original images, which provides an appropriate association for viewers. Moreover, our system only needs an image as input, so it can be used in some attractive applications, such as automatically generating corresponding visualization charts for different products on a website for promotion.

Fig. 8. Applications of charts embedded as elements into images and displayed as a whole.

Visualization charts are also widely used in presentation slide designs. In addition to images and charts, there are many other colorable components in presentation slides, and all of them will affect the final visual result. Different from the graphic design for posters or magazines, presentation slides, which can be produced quickly, are in high demand and are used more frequently by nonprofessionals. There are a large number of designed templates available on the internet for reference, but if users modify the content or images, they usually have to reselect the colors of other components accordingly. Moreover, a single presentation slide template cannot predefine visualization charts for different requirements, so users also need to design the chart and its palette themselves. Our system can solve the above problems. The palette generated using the image in the presentation slide can provide color references for both charts and other

Fig. 9. Two generated images using our system for presentation slides.

components. Two generated results using our system are shown in Figure 9. In addition to directly providing palette suggestions to users, our method can also cooperate with some existing automatic graphic design layout algorithms [65] to generate more personalized templates.

5.3 Color Design in Infographics

Infographics is a visualization form combining graphics and data content. It represents information efficiently with an artistic approach. As an essential element of infographics, color palettes greatly influence the aesthetic appeal, which is regarded by [46] as one of the criteria of users in selecting visualization products. Most of the current authoring tools used to create infographics, such as [66], only focus on providing layout suggestions. The predefined color palettes available within these tools limit the variety of choices and the possibility of modifying aesthetics in infographics.

The color design of infographics has always been challenging. Yuan et al. [67] combined deep learning with an interactive recommendation to generate color palettes for infographics. Users must set their color preferences during the generation process to obtain personalized results that satisfy some specific constraints, such as the preferred tonal style and semantic context. Cui et al. [68] proposed that a good infographic should have harmonious colors to help indicate latent semantics, and they proposed building a theme library to select semantically related colors predefined by the aesthetic experiences of users, which does not support customizing the visual style. These approaches demonstrate both the importance of visual aesthetic harmony and semantic contextual relevance in the palette design for an infographic.

Our method provides a new solution to address this issue. First, we customize palettes specifically for visualization, and the designed color selection strategy ensures that the results satisfy the perceptual discriminability requirements of infographics. There is no need for manual adjustment in the generation process. Second, as a special type of visualization, an infographic requires a harmonious color palette to achieve its unique artistry. The chroma and lightness of our results are adjusted to fit a linear harmony template, which has been proven in the art field to produce more harmonious color collections [61]. Third, compared with the limited predefined keywords to provide semantic information [68], massive images can indicate more diverse and precise semantics while carrying specific tonal styles selected by users.

Figure 10 shows several infographics produced by our system. Suppose the first two infographics have a white background and the last instance requires a recommended background color. Benefiting from the visual saliency detection network used in our method, we set the center text in (d) to be the most salient color (orange) of (a) and assigned the colors of the main subject in (b) (flowers and branches) to larger blocks of (e). Image (c) has cluttered subjects and large color contrast, and the light blue background is heavily occluded into small patches. Our method can still be used to identify and recommend the correct background color and extract a harmonious palette.

(a) Input (b) Input (c) Input

(d) Result (e) Result (f) Result

Fig. 10. Infographics instances. According to their original design templates, the color transparency of the bottom layer in (d) is 50%, and we assign the color with the highest saliency score in the palette to the center text. The darker colors (e) and (f) are generated by reducing the lightness of the original colors.

6 Evaluation

To demonstrate that our approach is suitable for visualization and can generate acceptable results, we performed three different evaluations, as follows: 1. two user experiments, where one used a questionnaire to measure data cognition accuracy for visualization, subjective preferences, and visual similarity of visualization instances and their original images;

the other is a user interaction experiment comparing other visualization palette generation tools and our system; 2. a reproducing experiment on designer-crafted palettes; and 3. some use cases for extreme situations that take "unsightly" images as input to show the practicality of displaying on black-and-white media.

6.1 User Experiment

6.1.1 User Research Questionnaire

In this experiment, we perform the evaluation of palettes from the following three aspects: *similarity*, where the similarity between the original image and generated color palette are compared; *aesthetics*, to determine whether the generated palette is aesthetically pleasing; and *separability*, to determine whether the palette sufficiently supports data cognition.

We designed a series of experiments with subtasks, including one discrimination task and five preference tasks. Each participant was asked to complete all these tasks. For the visualization of continuous and ordinal data, we sorted the colors of palettes generated by other methods by lightness. Because other methods cannot detect the background colors of images, we regarded the first color in the result as the background color when needed. In all the experiments in this section, "other methods" include the methods of Chang et al. [15], Adobe [14], and K-means [17]. Due to the number limitation of the Adobe approach, it was only used in tasks that need 5 colors.

We recruited 54 subjects, including 6 graphic design practitioners, 11 data visualization professionals, and 37 ordinary practitioners in other fields. The average age was 26.2, and there were two respondents with color weakness and blue-green blindness. We collected feedback results through questionnaires.

Similarity The predicted palette P should be similar to the original image I. This does not mean that each color in P should exist in I precisely, but the colors in P should provide a similar trend to I in terms of visual perception.

To prove that the color palette generated by our method is related to the input image, we evaluated the perceived similarity between the generated color palette and the original image by performing human-subject studies. In the first three tasks, we used other methods and our methods to generate palettes of an image and then asked subjects to choose the visualization instance most relevant to the image. The visualizations include a line stacking chart, pie chart, and infographic. There are 4, 3, and 4 available options in the three tasks, and the selection rates of our results are 48.15%, 75.93%, and 72.22%, respectively. We also selected three images with similar tones and used our method to generate three single-axis scatter diagrams. The participants were asked to match them to evaluate whether our method can accurately reflect the visual styles of different images. The rates for the three tasks providing correct results are 62.96%, 64.81%, and 75.93%. Finally, to measure the similarity gain of setting the correct background color and using our saliency-based color assignment strategy, we colored a diagram with the same palette using our method and two random methods in two tasks. Then, participants were asked to select the diagram most visually relevant to the original image. In both questions, 72.22% of the participants chose the diagram colored by our method. The above results demonstrate that our image-based color palette generation and colorization strategy can help the visualization effectively inherit the color visual style of the original image.

Aesthetics Aesthetics is performed as a significant global feature of the visualization. A well-designed theme color extraction method can inherit the aesthetics from the image. We performed two tasks to estimate the subject's aesthetic perception of the results. The first one is a single-factor experiment asking subjects to judge the pleasantness of the visualization instance generated by our method. The second task is a multifactor experiment to compare the pleasure felt when viewing the visualization instances generated by other methods and our methods.

For single-factor experiments, every visualization was attached to a $[-10, 10]$ range slider. Participants were asked to rate their colored visualization preference. Negative values indicate dislike, and positive values indicate preferences. We also provided zero as a neutral option. There are two scoring questions, and the average scores are 3.15 and 3.93. The multifactor experiment shows three or four visualization instances colored with palettes generated by different methods. Participants need to select the palette that performs best in terms of visual effect and rate it in the same way as the single-factor experiment. We designed three multifactor preference scoring questions and three single-choice questions with 4, 3, and 4 options. The (selection rate, average score) of our results in these questions are $(44.44\%, 5.71)$, $(48.15\%, 5.56)$, and $(46.3\%, 6)$, respectively, indicating that the acceptance of our method was higher, and our results were considered to have good aesthetic value. The detailed results are shown in Figure 11.

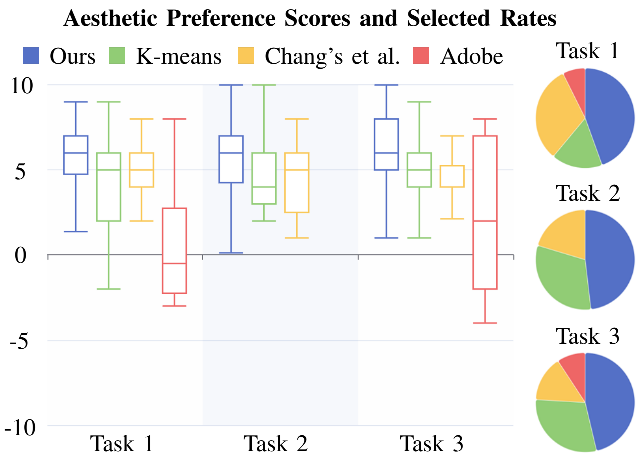


Fig. 11. The aesthetic preference scores and selected rates of three multifactor tasks.

Separability After encoding P to D and obtaining the final visualization, the colors for the categories of D need to be distinguishable. We used scatter plots for the discrimination experiments. We divided four rectangular regions in a 2D coordinate system and randomly generated 100 points in each area. The points were clustered by their coordinates into several groups, and we assigned colors to them. In this way, the demarcation of each group is relatively blurred and suitable for color-based discrimination tasks. To further increase the difficulty of identification, we set the transparency of all points in the scatter plots to 80%. The participants needed to identify the number of categories, which was used to calculate the absolute difference with the actual value to represent the error. We created two scatter plots with 5 categories. The identification accuracy of the participants was 94.44%, and the average error was 0.06. Both participants with color vision deficiencies answered correctly. The experiment shows that our method can generate distinguishable colors for visualizations.

6.1.2 User Interaction Experiment

Some existing tools support the generation of unique visualization palettes through user interaction, such as Colorgorical [20] and Palettailor [21]. Inspired by these systems, we developed a customized interface based on our method, which allows users to input an image and generate a palette. It supports the number of colors from 4 to 7 and provides several chart instances for users to preview the coloring effect. This system contains the core functionality of our method, and we compared it with Colorgorial and Palettailor in this experiment.

We recruited five participants with little experience in visualization design, including 3 males and 2 females. They were asked to complete the following two tasks in order using Adobe Photoshop Color Picker (designing palette by themselves), Colorgorical, Palettailor, and our system.

(1) Generate a personally satisfying 6-color visualization palette with a white background.
(2) Generate a personally satisfying 6-color visualization palette with one additional background color. Moreover, its color style should differ from the first generated palette.

Before the experiment, for each tool, we spent approximately five minutes explaining to the participants how to use it. When using our system, the participants can freely select input images from the internet or the provided folder with 100 natural images. When the participants performed task 2 using Colorgorical and Palettailor, they had to generate a 7-color palette first and then select a color from it as the background. We recorded the time spent on each task and then asked the participants to rate the ease of use of these tools after the experiment. Then, for each participant, we assigned the colors of their result palettes to two charts and asked them to rate their satisfaction. Finally, we interviewed them and asked about their feelings regarding using these tools. The detailed data and the generated visualization instances are provided in the Appendix for reference.

Time Spent The time spent in (task1, task2) of different tools are as follows: Adobe Photoshop Color Picker (79s$\pm$57s, 97 s$\pm$43s), Colorgorical (108s$\pm$60s, 97 s$\pm$43s), Palettailor (96s$\pm$19s, 125 sp474/¿48s), and our system (50s$\pm$14s, 33 s$\pm$19s). The generation efficiency values of different users vary greatly, resulting in large standard deviations in the time cost data. It can be observed that the participants took the least time to obtain a satisfactory palette using our system, especially in task 2. Participant 1 commented, "When using Colorgorical, if I'm not satisfied with a result, I have to retune the parameters. However, in fact, I'm not sure about the use of some parameters...The next result is somewhat random for me."

Ease of Use The participants subjectively rated the ease of use of these tools on a five-point scale (Colorgorical: 3$\pm$0.89, Palettailor: 3.2$\pm$0.75, our system: 4.2$\pm$0.4). Our system received the highest score, possibly because users only have to set the size of the palette, and it is easy for them to choose images with harmonious colors. For nonprofessional users, some of the parameters in Colorgorical and Palettailor may be difficult to understand or utilize. Participant 1 mentioned that "The Colorgorical and Palettailor provide parameters in multiple aspects, but they lack recommendations. It is hard for me to tell which parameters are more important...When browsing and selecting images, I feel relaxed and delighted." Participant 4 said our system is "like a point and shoot camera that everyone can use."

Satisfaction For each participant, we colored 4 donut charts and 4 rose charts using palettes they generated in two tasks and asked them to rate their level of satisfaction regarding the charts on a five-point scale. In task 1, the palettes the users designed themselves received the highest satisfaction scores, followed by those generated by our method (Adobe Photoshop Color Picker: 3.6$\pm$0.8, our system: 3.5$\pm$0.8, Colorgorical: 2.4$\pm$0.49, Palettailor: 2.2$\pm$0.4). In task 2, our method received the highest scores, which are higher than those in task 1 (our system: 4.5$\pm$0.49, Adobe Photoshop Color Picker: 3.2$\pm$0.4, Colorgorical: 3$\pm$1.4, Palettailor: 2.2$\pm$0.98). Some of the background colors specified by the participants are not appropriate for the generated charts. Participant 2 said, "It is likely to choose an inappropriate background color if I cannot preview the chart. However, I can infer from an image whether its background is a good match."

6.2 Reproducing Experiment

Compared with other image-based palette extraction work, our method achieves finer-grained color clustering in salient regions, which enables the result to contain some important colors even though the colors occupy small areas in the original image. This feature is similar to the design patterns of professional when extracting palettes from images. In addition, we adjusted the results according to color harmony principles to ensure that the final palette is more in line with the aesthetic trends of professional designers.

We collected 200 high-quality works that contain paired images and palettes produced by professional designers from the internet. Then, we compared the results from our method with the results generated by the K-means method [17] and the method proposed by Chang et al. [15] and to the target results designed by the designer. We calculated the distance between two palettes using the minimum color difference model [69] to evaluate their similarity.

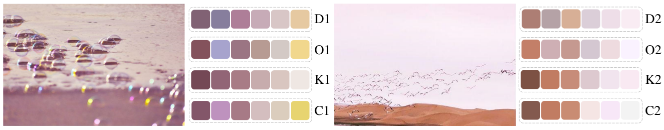


Fig. 12. The palettes produced by professional designers (D), our method (O), K-means (K), and the improved color clustering algorithm of Change et al.(C).

Figure 12 shows two examples. It is worth noting in the examples that our method can extract some small but important colors, such as blue-violet and yellow, which are reflected by the foam in the left example. Our palette in the right example has a softer lightness variation compared to the other methods. A detailed comparison of palette distance is shown in Table 1. Our results are more similar to the designer's choices in general. However, there are still 16 palettes that differ greatly from the designer's results, as shown in the Appendix. After analyzing them we found two main reasons for this difference, as follows: (1) our method extracted more colors from the foreground of the image, while the designer selected colors more evenly form these images; and (2) when the image contains many neutral colors, the designer's palettes tend to be colorful, while ours are not because our method does not prioritize colorful colors.

TABLE 1
The palette distance between the results of previously described methods and that generated by designers.

Method	K-means	Chang et al.	**Our**
Average palette distance	196.88	196.04	**186.60**
Number of similar instances (Palette Distance ≤ 180)	14	14	**45**
Number of dissimilar instances (Palette Distance ≥ 200)	74	74	**16**

6.3 Color Generation for Extreme Cases

The experiments in this section consider some extreme input images and display platforms of visualization. The results show that our method can also perform well in these special situations.

6.3.1 Input Images with Poor Quality

Our method is image-based, so the quality of the results will inevitably depend on the input image to a certain extent. Images with high aesthetic quality are more likely to generate excellent results, but users may select inappropriate images as input. We analyzed some of the worst situations and divided the discussion into two situations. In the first situation, the aesthetic quality of the image itself is high, but the image quality is poor, which may result from a large amount of noise or low resolution. In the second situation, the image is of low aesthetic quality, and good aesthetic perception cannot be achieved.

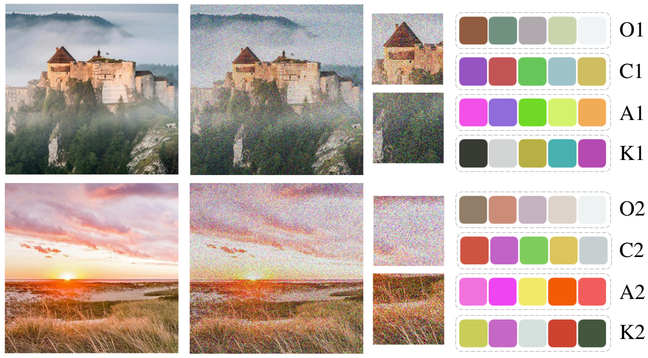


Fig. 13. Each image set from left to right shows the original image, the image with 30% multicolor noise, the details in the noisy image, and the resulting palettes generated by our (O), Chang et al.(C), Adobe's(A) and K-means (K) methods.

For the first situation, we added noise to the original image and compared the difference between the results generated by our method and the K-Means, Chang et al., and Adobe's methods. Specifically, we used the noise addition function in the Adobe Photoshop filter tool to add 30% of the multicolor noise in a Gaussian distribution. For the processed image, the visual perception is not obviously changed due to the darker background color tone. We used the methods mentioned above to generate the color palette. As shown in Figure 13, the palettes extracted by other methods are largely affected by noise, and our method overcomes this negative influence.

The second situation is that we selected some "ugly" images as input. These "ugly" input images come from the dataset AVA [70]. AVA is an aesthetic evaluation dataset in which each image contains a score rating from approximately 200 individuals, and the score ranges from $[1, 10]$. The scorers include not only ordinary image enthusiasts but also professional designers, so the scoring is universal to measure the image aesthetic quality. The higher the score is, the higher the aesthetic quality of the image. We calculated a weighted average of these scores to obtain the aesthetic score corresponding to each image. We selected some images with scores below 3, which means that more people perceive this image as "ugly". The results are shown in Figure 14. We can see that even with a poor input, our model can make certain adjustments to the output colors to make the final results pleasant.

6.3.2 Generating Palette for Special Display Platforms

Currently, low-power e-ink screens are widely used, and some paper media, such as newspapers and books, are printed in black and white. To optimize the display effect of visualization works on these special platforms and improve the reading efficiency of color-weak or color-blind people, we tried to

Fig. 14. Palettes generated with low aesthetic quality images.

ensure that the Bavarian distance (Equation 10) between any two CIELCh colors in lightness and chroma is greater than 3, which helps the colors remain distinguishable even in grayscale.

TABLE 2
We converted the colors extracted by different methods to grayscale and calculated the difference between the two closest colors in each palette.

Method	K-means	Chang et al.	Adobe	**Our**
Average gray difference	25.23	20.31	10.70	**22.65**
Instance number with good discriminability (Gray difference ≥ 25)	93	67	14	**78**
Instance number with poor discriminability (Gray difference ≤ 10)	36	68	114	**16**

We designed an experiment that qualitatively evaluates the discriminability of color palettes in grayscale space. For an RGB color (r,g,b) $(r \in [0,255], g \in [0,255], b \in [0,255])$, we converted the color to a single-channel grayscale color *gray* by $gray = 0.299*r+0.587*g+0.114*b$. After conversion, we examined the distance between the two closest colors in the palette. We crawled 200 images from the internet and used several image-based color extraction methods to generate results; see Table 2 for details. The methods of Chang et al. and K-means focus on colors with large differences in the image, while Adobe's method is more similar to ours, as it also considers the harmony and aesthetic characteristics of the result. Therefore, the palettes generated from the K-means methods perform better than our method in this experiment. Adobe's method performs the worst among all methods. Although our average result in this experiment is not as good as that of the K-means method, the number of instances with poor discriminability is less, benefiting from our selection strategy, which avoids colors with very similar lightness and chroma values.

7 Limitations and Discussion

Our current method still has some limitations. First, the number of colors in the generated palette can be specified in our method, but the recommended range is $4 \sim 7$. If the specified color number is too high $(n \geq 10)$, similar colors tend to appear in the results for most input images, which may affect the separability of categories in visualization instances. If the number of colors is too small, our selection strategy may ignore some areas with lower visual significance of the image. Our selection approach can be further improved to strike a balance between inheriting the overall visual style of the image and extracting colors with high visual saliency.

Second, we have designed different color assignment strategies for multitype visualizations, but limitations on the input images still exist. For discrete data, if the input image presents a narrow color range, the number of colors that meet the requirements in the selection process may not be sufficient, which can result in an undesirable palette. For visualizations of continuous data, we do not suggest inputting images with complicated colors. If the input image contains multiple tones, the result with two main colors might not well inherit its visual style. We will work on designing more flexible color encoding strategies to accommodate various visualization cases better in subsequent research.

Third, the visual style of our result only depends on the input image and cannot be styled on the same input. For example, users cannot specify the style of the results to be **colorful**, **bright** or **muted** after inputting an image.

8 Conclusion

Images provide rich color inspiration, carrying semantic information that can inspire associations. The image-driven approach is a feasible and effective way to generate a harmonious and context-aware color palette. We have proposed an image-based approach to generate harmonious color palettes for visualization. We put forward a unique color clustering method based on visual importance and hue value, which enables more fine-grained color extraction in the subject area than in the background area of an image. The subsequent color selection and optimization can generate harmonious visualization palettes. We recommend different color combinations according to the different context forms of the visualization and design two color optimization and assignment strategies for visualizations of different data types to encode colors to data in visualizations. Through evaluation, we demonstrated the usability of our method in visualization tasks and the harmony, stability, and wide applicability of the generated palette. We hope that this work can provide a reference for visual color design, stimulate design inspiration, and provide convenience in use for people of different skill levels. Our code is open-source at https://github.com/Shuqi-67/ColorPipette.

References

[1] D. Borland and R. M. T. Ii, "Rainbow color map (still) considered harmful," *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 14–17, 2007.
[2] Q. Zeng, Y. Wang, J. Zhang, W. Zhang, C. Tu, I. Viola, and Y. Wang, "Data-driven colormap optimization for 2d scalar field visualization," in *2019 IEEE Visualization Conference (VIS)*, 2019, pp. 266–270.
[3] N. Zhao, Y. Cao, and R. W. Lau, "What characterizes personalities of graphic designs?" *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–15, 2018.
[4] K. E. Burchett, "Color harmony," *Color Research & Application*, vol. 27, no. 1, pp. 28–31, 2002.

[5] Y. Matsuda, "Color design," *Asakura Shoten*, vol. 2, no. 4, p. 10, 1995.

[6] M. Tokumaru, N. Muranaka, and S. Imanishi, "Color design support system considering color harmony," in *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, vol. 1. IEEE, 2002, pp. 378–383.

[7] L.-C. Ou, M. R. Luo, A. Woodcock, and A. Wright, "A study of colour emotion and colour preference. part iii: Colour preference modeling," *Color Research & Application*, vol. 29, no. 5, pp. 381–389, 2004.

[8] N. Kita and K. Miyata, "Aesthetic rating and color suggestion for color palettes," in *Computer Graphics Forum*, vol. 35, no. 7. Wiley Online Library, 2016, pp. 127–136.

[9] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu, "Color harmonization," in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 624–630.

[10] Y. Shi, P. Liu, S. Chen, M. Sun, and N. Cao, "Supporting expressive and faithful pictorial visualization design with visual style transfer," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[11] Q. Zhang, C. Xiao, H. Sun, and F. Tang, "Palette-based image recoloring using color decomposition optimization," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1952–1964, 2017.

[12] S. Kim and S. Choi, "Automatic color scheme extraction from movies," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020, pp. 154–163.

[13] P. Weingerl, A. Hladnik, and D. Javoršek, "Development of a machine learning model for extracting image prominent colors," *Color Research & Application*, vol. 45, no. 3, pp. 409–426, 2020.

[14] "Adobe color," https://color.adobe.com/zh/create/image Accessed March 1, 2022.

[15] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring." *ACM Trans. Graph.*, vol. 34, no. 4, pp. 139–1, 2015.

[16] Y. Aksoy, T. O. Aydin, A. Smolić, and M. Pollefeys, "Unmixing-based soft color segmentation for image manipulation," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 2, pp. 1–19, 2017.

[17] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[18] J. Tan, J.-M. Lien, and Y. Gingold, "Decomposing images into layers via rgb-space geometry," *ACM Trans. Graph.*, vol. 36, no. 1, pp. 7:1–7:14, 2016.

[19] T. Gerstner, D. DeCarlo, M. Alexa, A. Finkelstein, Y. Gingold, and A. Nealen, "Pixelated image abstraction," in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, ser. NPAR '12, 2012, p. 29–36.

[20] C. C. Gramazio, D. H. Laidlaw, and K. B. Schloss, "Colorgorical: Creating discriminable and preferable color palettes for information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 521–530, 2016.

[21] K. Lu, M. Feng, X. Chen, M. Sedlmair, O. Deussen, D. Lischinski, Z. Cheng, and Y. Wang, "Palettailor: Discriminable colorization for categorical data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 475–484, 2020.

[22] S. Smart, K. Wu, and D. A. Szafir, "Color crafting: Automating the construction of designer quality color ramps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1215–1225, 2019.

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[24] Y. Huang, C. Wang, and C. Li, "Translucent image recoloring through homography estimation," *Computer Graphics Forum*, vol. 37, no. 7, pp. 421–432, 2018.

[25] E. Kim and H.-J. Suk, "Key color generation for affective multimedia production: An initial method and its application," in *Proceedings of the 24th ACM International Conference on Multimedia*, ser. MM '16, 2016, p. 1316–1325.

[26] K. S. Tan and N. A. M. Isa, "Color image segmentation using histogram thresholding–fuzzy c-means hybrid approach," *Pattern Recognition*, vol. 44, no. 1, pp. 1–15, 2011.

[27] Z.-J. Du, K.-X. Lei, K. Xu, J. Tan, and Y. Gingold, "Video recoloring via spatial-temporal geometric palettes," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–16, 2021.

[28] P. O. Donovan, A. Agarwala, and A. Hertzmann, "Color compatibility from large datasets," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, pp. 1–12, 2011.

[29] Z. Feng, W. Yuan, C. Fu, J. Lei, and M. Song, "Finding intrinsic color themes in images with human visual perception," *Neurocomputing*, vol. 273, pp. 395–402, 2018.

[30] D. Borland and A. Huber, "Collaboration-specific color-map design," *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 7–11, 2011.

[31] L. Yuan, W. Zeng, S. Fu, Z. Zeng, H. Li, C.-W. Fu, and H. Qu, "Deep colormap extraction from visualizations," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2021.

[32] C. G. Healey, "Choosing effective colours for data visualization," in *Proceedings of Seventh Annual IEEE Visualization'96*. IEEE, 1996, pp. 263–270.

[33] E. R. Tufte, N. H. Goeler, and R. Benson, *Envisioning information*. Graphics press Cheshire, CT, 1990, vol. 2.

[34] B. A. Maxwell, "Visualizing geographic classifications using color," *The Cartographic Journal*, vol. 37, no. 2, pp. 93–99, 2000.

[35] L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller, "Color design for illustrative visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1739–1754, 2008.

[36] J. Ahmad, E. Huynh, and F. Chevalier, "When red means good, bad, or canada: Exploring people's reasoning for choosing color palettes," in *2021 IEEE Visualization Conference (VIS)*. IEEE, 2021, pp. 56–60.

[37] F. Samsel, G. Abram, S. Zeller, and D. Keefe, "Affective palettes for scientific visualization: Grounding environmental data in the natural world," in *2021 IEEE VIS Arts Program (VISAP)*. IEEE, 2021, pp. 20–34.

[38] S. Lee, M. Sips, and H.-P. Seidel, "Perceptually driven visibility optimization for categorical data visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 10, pp. 1746–1757, 2012.

[39] C. A. Brewer, "Color use guidelines for mapping," *Visualization in Modern Cartography*, vol. 1994, pp. 123–148, 1994.

[40] B. E. Rogowitz and A. D. Kalvin, "The" which blair project": a quick visual method for evaluating perceptual color maps," in *Proceedings Visualization, 2001. VIS'01.* IEEE, 2001, pp. 183–556.

[41] C. Ware, "Color sequences for univariate maps: Theory, experiments and principles," *IEEE Computer Graphics and Applications*, vol. 8, no. 5, pp. 41–49, 1988.

[42] C. Tominski, G. Fuchs, and H. Schumann, "Task-driven color coding," in *2008 12th International Conference Information Visualisation*, 2008, pp. 373–380.

[43] A. Zeileis, K. Hornik, and P. Murrell, "Escaping rgbland: Selecting colors for statistical graphics," *Computational Statistics & Data Analysis*, vol. 53, no. 9, pp. 3259–3270, 2009.

[44] M. Tennekes and E. de Jonge, "Tree colors: color schemes for tree-structured data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2072–2081, 2014.

[45] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, 2016.

[46] K. Moreland, "Diverging color maps for scientific visualization," in *International Symposium on Visual Computing*. Springer, 2009, pp. 92–103.

[47] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3166–3173.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[49] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 478–487.

[50] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7479–7489.

[51] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

[52] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

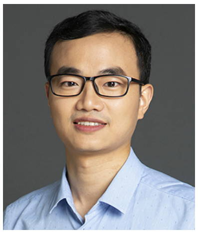
[53] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Learning layouts for single-pagegraphic designs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 8, pp. 1200–1213, 2014.
[54] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Computer Vision, IEEE International Conference on*, vol. 2. IEEE Computer Society, 2003, pp. 10–10.
[55] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
[56] W.-C. Tu, M.-Y. Liu, V. Jampani, D. Sun, S.-Y. Chien, M.-H. Yang, and J. Kautz, "Learning superpixels with segmentation-aware affinity loss," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 568–576.
[57] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 964–13 973.
[58] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2010.
[59] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.
[60] X. Wang and J. Yu, "Learning to cartoonize using white-box cartoon representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8090–8099.
[61] C. Lara-Alvarez and T. Reyes, "A geometric approach to harmonic color palette design," *Color Research & Application*, vol. 44, no. 1, pp. 106–114, 2019.
[62] G. Sharma, W. Wu, and E. N. Dalal, "The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research & Application*, vol. 30, no. 1, pp. 21–30, 2005.
[63] C. A. Brewer, "Designing better maps," *A Guide for GIS Users*, vol. 1, 2005.
[64] "Aliyun luban color-recog," https://luban.aliyun.com/web/ai-lab/color-recog Accessed March 1, 2022.
[65] X. Zheng, X. Qiao, Y. Cao, and R. W. Lau, "Content-aware generative modeling of graphic design layouts," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–15, 2019.
[66] H. Xia, N. Henry Riche, F. Chevalier, B. De Araujo, and D. Wigdor, "Dataink: Direct and creative data-oriented drawing," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–13.
[67] L.-P. Yuan, Z. Zhou, J. Zhao, Y. Guo, F. Du, and H. Qu, "Infocolorizer: Interactive recommendation of color palettes for infographics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4252–4266, 2021.
[68] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang, "Text-to-viz: Automatic generation of infographics from proportion-related natural language statements," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 906–916, 2019.
[69] Q. Pan and S. Westland, "Comparative evaluation of color differences between color palettes," in *Color and Imaging Conference*, vol. 2018, no. 1. Society for Imaging Science and Technology, 2018, pp. 110–115.
[70] N. Murray, L. Marchesotti, and F. Perronnin, "Ava: A large-scale database for aesthetic visual analysis," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2408–2415.

Shuqi Liu received his B.Eng. from East China Normal University, China, in 2022. She is working toward the Master degree with East China Normal University, Shanghai, China. Her main research interests include information visualization and graphics design.

Mingtian Tao received his B.Eng. from East China Normal University, China, in 2020. She is working toward the Master degree with East China Normal University, Shanghai, China. Her main research interests include computer graphics and image analysis.

Yifei Huang received her Ph.D. from East China Normal University, China, in 2022. His main research interests include computer graphics and image analysis.

Changbo Wang is a professor with the School of Computer Science and Technology, East China Normal University. He received his Ph.D. degree at the State Key Lab of CADCG of Zhejiang University in 2006. He was a postdoctor of the State University of New York in 2010. His research interests mainly include computer graphics, information visualization, visual Analytics, etc.



Chenhui Li received Ph.D. from the Department of Computing at Hong Kong Polytechnic University, in 2018. He is an associate professor with the School of Computer Science and Technology at East China Normal University. He received ICCI*CC Best Paper Award (2015) and SIGGRAPH Asia Sym. Vis. Best Paper Award (2017). He has served as a local chair in VINCI2019. He works on the research of information visualization and computer graphics.