Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

Towards Natural Object-based Image Recoloring

Meng-Yao Cui[1], **Zhe Zhu**[2], **Yulu Yang**[1], **Shao-Ping Lu**[1] (✉)

Abstract Existing color editing algorithms enable users to edit the colors in the image according to their own aesthetics. Unlike artists who have an accurate grasp of colors, ordinary users are inexperienced in color selection and matching, and allowing non-professional users to edit the colors arbitrarily would lead to unrealistic editing results. To address this issue, we introduce a palette-based approach for realistic object-level image recoloring. Our data-driven approach consists of an offline learning part that learns the color distribution of different objects in the real world, and an online recoloring part that first recognizes the object categories, then recommends realistic candidate colors learned in the offline step according to the objects category. We also provide an intuitive user interface for efficient color manipulations. After the color selection, image matting is performed to ensure the smoothness of the object boundary. Comprehensive evaluations on various color editing examples demonstrate that our approach outperforms existing state-of-the-art color editing algorithms.

Keywords color editing, object recognition, color palette representation, natural color.

1 Introduction

Manipulating the color of an image is a fascinating work which draws a widespread attention. By changing the color, we can change theme [39], style [21], illumination [12] and even emotion [17] of pictures. In terms of methodology there are several ways to recolor an image. One is to map the color from the source image to the target image, which is also called color transfer. The mapping process can be either based on geometry [50, 52] or the statistics [22, 23]. From the user interaction perspective, stroke-based recoloring and palette-based recoloring are two popular color manipulation methods. In stroke-based recoloring such as edit propagation [2, 55], users only provide sparse inputs while the algorithm propagates the edits to the proper regions in the rest of the image, based on the pixel-level affinities. To further alleviate the user interaction burden, palette-based coloring [5] allows users to recolor an image by just editing a color palette. This is intuitive and users can adjust the color based on instant feedback. Previous palette-based recoloring methods mainly focus on the generation of source palette [5, 49, 59] and the transfer from source color to target color [8]. They assume that target color is given or chosen according to the preference of users, which works for artists. For unprofessional users who do not have an accurate grasp of colors, they cannot select the exact color of a particular object, since they can distinguish between green and red but have no sense of the subtle differences between magenta and carmine. It is also time-consuming for them to consider the relationship between those colors of the palette.

Allowing users to edit the color arbitrarily can give rise to unrealistic editing results. One way to solve this problem is

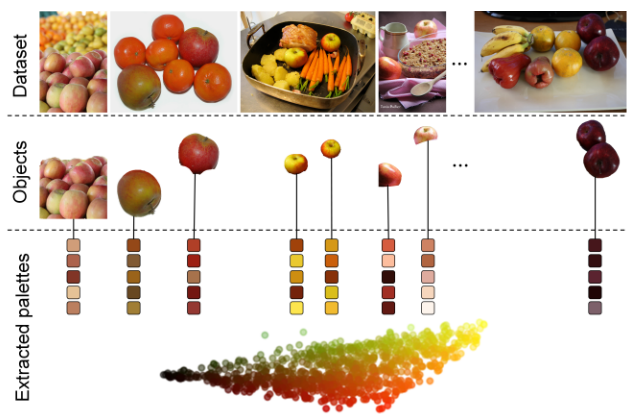

Fig. 1 An example of the palette extraction process of our work. For the category "apple", we first extract apples from the images, then calculate the palettes of them separately.

1 TKLNDST, CS, Nankai University, Tianjin, China. Email: imclimmy@qq.com; slu@nankai.edu.cn (✉).
2 Department of Radiology, Duke University.
Manuscript received: 20xx-xx-xx; accepted: 20xx-xx-xx.

to use a reference picture [17] to generate the target palette. This approach is more efficient than directly choosing the color. However, finding a proper reference picture is still a challenging task for users. There are also several colorization methods which provide diverse and realistic colorization results [11, 43], but their randomly generated results are not flexible for users to choose. Our solution of this problem is based on the following observations. a) The color of objects in nature varies within a limited range, and the color of most objects can be represented by a number of representative colors. For example, the color of carrot is always between yellow and red while zebras have black and white stripes; b) Due to its large scale and diversity, in image dataset such as Microsoft COCO [30], the color distribution of an object category is already able to represent its actual color distribution in the real world. Thus, the key idea of this paper is to adopt a data-driven approach that learns the color palette of object categories from large-scale image dataset in advance, and recommend the target color palette based on the recognized object categories and pre-learned models. We propose a novel image recoloring approach which can recommend the target palette based on the object and its corresponding color pattern in the image dataset. Our approach consists of an offline palette generation part and an online recoloring part. In the offline part, we generate color palettes for different objects using the provided masks (see an example of apples in Fig. 1). After that, we merge the palettes within the same object category and provide a default palette for each object category. Once the palettes are generated, we can apply them to our online recoloring approach. For an input image, we detect the objects in the image, and generate the alpha mattes with the detected masks. After that, we transfer the colors of the foreground using the recommended palettes and blend it with background. Our framework can also be applied on global color editing to meet more color editing requirements. Finally, a user interface is further provided, in which users can edit the image intuitively and efficiently. our main contributions are summarized as follows:

- We propose a novel natural recoloring approach which recommends varied target palette based on the color distribution of real objects.
- We provide an intuitive and efficient user interface for color manipulation.

2 Related work

Our work is related to palette-based color manipulation, semantic based color transfer, matting and color editing. In the following we review the most relative works from those aspects.

Palette based color manipulation. Palette based color manipulation edit the image by modifying the palettes. Those approaches can be divided into two groups. One group takes use of a pair of palettes for each image, the source palette and the target palette. A main way of generating the source palette is clustering the colors in an image using k-means method [5, 41, 59], which rely on the global color distribution. Kang *et al.* [22] provide an approach which can capture the local distinctive colors. Furthermore, the method based on RGB values are not enough to recover all features of paintings. Thus, several works are designed to extract the original pigments in the paintings, which contains different physical properties [1, 49]. Some methods further decompose the image into several mixing layers which corresponding to the colors in the palette separately [49, 50]. Huang *et al.* [18] also provide a method which is designed for transparent objects recoloring. Another group only needs target palette, which is widely used in color compatibility [40], theme enhancement [29, 51] and recolorization [8]. This group of methods can also be used in colorization tasks. For instance, Bahng *et al.* [4] colorize the picture using the palette generated by text. Other applications such as transferring color theme [39] and expressing different emotions [17] also implemented with palette-based method.

Semantic based color transfer. Color transfer methods recolor the image by taking one or more images as reference, which can be used in both of colorization and recolorization. Many methods of color transfer base on the statistics of color distribution [22, 23, 42], while there are also geometry-based methods [50, 52] and user-assisted solutions [13]. During color mapping, multi-level semantic information is considered. Welsh *et al.* [53] match luminance and texture between images and transfer chromatic information. The method proposed in [20] uses the higher-level context of the pixels to transfer colors automatically, which is implemented by a supervised classification scheme. Super-pixels are then matched between images to achieve higher extent of spatial consistency [14]. Furthermore, Iizuka *et al.* [19] combine global and local image features using deep network, freeing the users from providing reference images. There are also several methods take use of object-level information. The framework designed by Ma *et al.* [38] learns the object-level correspondences for image translation. In colorization, colorizing instance and background separately and merging the results with a fusion module improves the performance [46]. Facial skin beautification [28] also takes use of generated face masks.

Matting. Matting provides multiple utilities to separate the object from its background. Traditional matting methods try to solve the matting equation using trimap or scribbles [7, 9, 25, 47]. The results of these methods highly depend on the

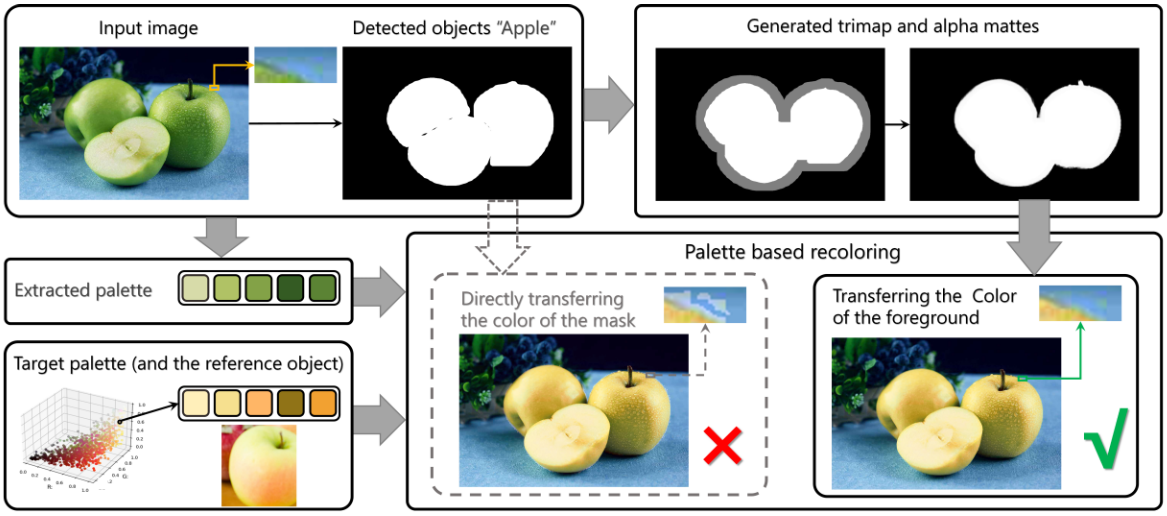

Fig. 2 Pipeline of our work. We detect the object in the image and generate the palette based on the mask. We use alpha matting to process the mask and get the α matte of the object. Finally, we transfer the color of foreground using the target palette chosen by users.

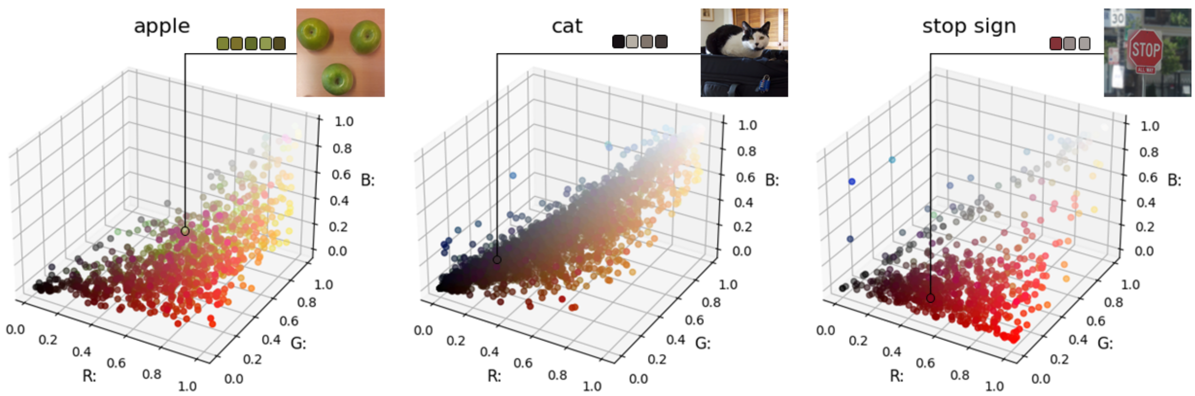

Fig. 3 Samples of the generated palettes. For each palette, only one color with the most adjacent colors in the object is shown. The R, G, B values in the diagram are linearly normalized to range $[0,1]$. We also show the center palettes of each category with their corresponding images.

color cues. Recent years, matting methods based on deep network has been largely developed. A convolutional neural network [45] is designed which considers both semantic prediction and matte optimization. The dataset they used are generated by traditional matting method. Xu *et al.* [57] design a two-parts model, in the first part they predict mattes from an image and its trimap, then they refine the alpha mattes in the second part. To solve human matting problems, Wu *et al.* [54] further proposes an end-to-end learning framework which combines pose estimation with trimap and matting network. Especially, Lu *et al.* [31] put up with a flexible network which can perform well in natural image matting tasks.

Color editing. Multiple tools have been designed for users to implement color editing. One kind of color edit methods allow users to edit the color of all pixels in the image, which always take scribbles such as color points or strokes [24, 36] as input. Those methods are later improved from the aspect of running time [55, 56] and the consumed labor [60]. Another kind of methods implement higher-level of image editing, for instance, some methods take reference image to assist the editing. In the framework provided by An *et al.* [3], users draw pairs of strokes in target and reference images, which indicate the regions with the same color "style", the reference colors is transferred to the target images based on the stroke pair. Lu *et al.* [34] design a color retargeting approach, which composes the time-varying color image. Besides that, color editing has been applied in multi-view [32, 33, 35, 58] and 3D analysis [10, 26]. Some researchers also attempt to preserve the aesthetic or creative intentions during transferring [16, 27, 48]. Palette-based recoloring is another popular way used for color editing, and as stated before, users can edit the image by modifying the colors in the palette. Most of works focus on the extraction of original palette and the effect of recoloring, think less of the selection of the target palette, only several methods provide the recommend palettes for users. Such recommend can be produced based on the predicted color distribution [60] or online repositories [59]. Huang *et al.* [17] establish a dataset of palettes with different emotions. For each picture, palettes of different emotion can be recommended based on its original palette. Different from these methods, our approach aims at implementing natural and realistic recoloring with recommending palettes.

3 Approach

Our approach consists of an offline candidate palettes generation step and an online recoloring step. In the offline step we first generate thousands of initial palettes, then merge similar palettes to obtain representative palettes for each object category. This color model can be saved and used in the online recoloring step, where given an input image to recolor, we detect and segment the objects, and based on the recognized category of objects we provide candidate palettes for users. The pipeline of our work is shown in Fig. 2. Next, we give the details of those steps.

3.1 Palette generation

We use the images in Microsoft COCO dataset [30] to generate the candidate palette for each object category in the offline step, since this dataset is of large scale and it also provides the mask of the objects of 80 classes. In the online recoloring, we use the Mask-RCNN [15] trained on COCO dataset to extract the mask of the objects of the input image. The network is built with Open MMLab Detection [6]. Given that the segmentation output of deep neural networks is always rough and inaccurate along the boundary, we use alpha mattes for further refinement. To generate the trimap for alpha mattes, we apply corrosion-expansion operation on the segmentation results. In detail, the original segmented masks are corroded and expanded for 15 pixels separately. The region between the corroded mask and expanded mask is regarded as unknown region, the corroded mask is regarded as foreground region, and the rest of the picture is background. The trimap is then used to generate the alpha mattes by Indexnet [31]. This refinement is performed in both offline and online step. An example of the improved segmentation result using alpha mattes is illustrated in Fig. 2.

3.1.1 Initial palette extraction

We use the method proposed by Chang *et al.* [5] to extract the initial palettes since it can be performed in real time. We extract the palettes of 80 categories using the masks of the images of the dataset. More specifically, we first generate the palette for each masked region. In [5], the number of the colors in the palette (denoted by k in this paper) should be given beforehand. Instead of setting a fixed k for all the palette, we adopt an adaptive strategy by determining k of each category based on the following calculation. We first generate palettes with $k = 3,4,5,6$ (a larger k value makes the generation speed slower) separately for all masked regions and calculate the loss of each palette using the following equations:

$$\begin{cases} d_i = dist(p_i, C_i), \\ loss_p = ||d||_2^2, \end{cases} \tag{1}$$

where p_i is the color of $i-$th pixel of the masked region, C_i is a color of palettes corresponding to p_i. $dist(\cdot,\cdot)$ calculates the color difference between two pixels using CIEDE2000, which is a closer metric to human assessment than RGB Euclidean distance [37]. After that, we calculate the ratio for each loss of the palettes as follows:

$$\begin{cases} D_i = dist(p_i, C), \\ ratio = \frac{loss_p}{||D||_2^2}. \end{cases} \tag{2}$$

In this equation, C is the mean color of all pixels in the mask. For each category, we calculate the mean ratio with $k = 3,4,5,6$ separately, and choose the smallest number of the colors whose corresponding mean ratio is less than 0.20. Finally, we collect the palettes with k representative colors for each kind of objects. Fig. 3 shows some examples of the generated palettes. Note that some categories with too uncertain color patterns are manually ruled out, for example, ”person”, ”handbag” and ”bottle”.

3.1.2 Palette merging

The initial palette generation step usually produces thousands of palettes, many of which have similar colors. To provide tens of candidate palettes for interactive recoloring, similar palettes are thus merged to get representative palettes. We use density-based spatial clustering of applications with noise method (DBSCAN) [44] to merge the palettes (see some results in Fig. 4).

We also need a recommended palette for each category. To implement this, we set the distribution center of all the palettes for each category, which has the least sum of distance (Sd) comparing with other palettes, as the default palette for recoloring. For each palette x, the distance between it and any other palette r_i, which is the i-th palette of the corresponding category of x, is calculated as follows:

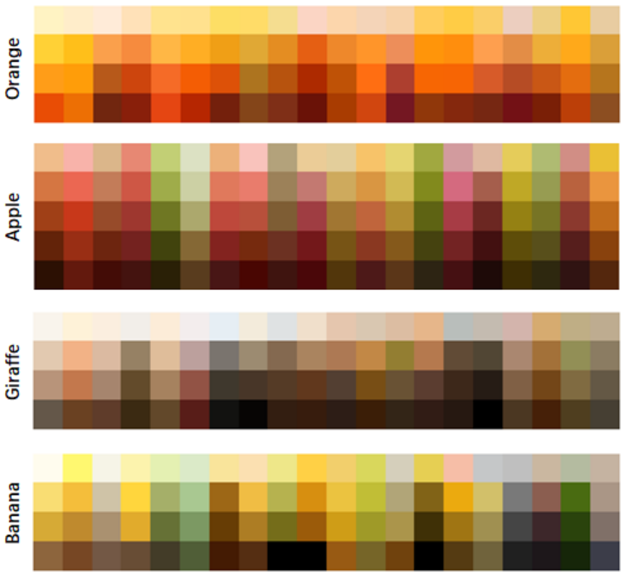

Fig. 4 Some examples of recommending palettes. In each category, a column of colors corresponds to a recommend palette, twenty palettes are shown for each category.

Fig. 5 Some transferring results for images with multi-objects.

$$\begin{cases} d_j = dist(x_j, r_{ij}), \\ pd_i = ||d||_2^2, \end{cases} \tag{3}$$

where x_i represents the i-th color of palette x, r_{ij} denotes the i-th color of palette r, and pd_i is the distance between x and r_i. The colors in x and r_i are ranked by L values in the LAB color space. Sd is then calculated using the following equation:

$$Sd = \sum_{i}^{n} pd_i, \tag{4}$$

where n is the number of palettes in the corresponding category of x. Therefore, the distribution center of the palettes is the default palette used in recoloring (see Fig. 3).

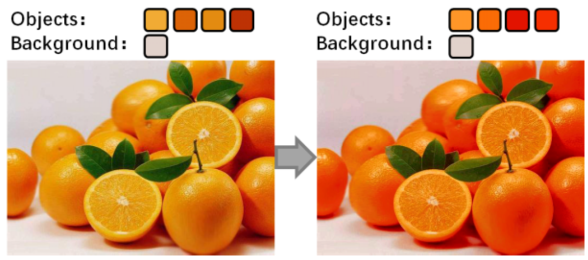

Fig. 6 An example for global recoloring. We extract five colors from the objects and a color from the background. The five colors of the objects are transferred while the colors of the background are preserved.

3.2 Instance-based recoloring

In the online recoloring step, given an input image to recolor, we use the original palette extracted from the object region and the recommend palette to implement instance-based recoloring. During color transfer, each color in the original palette needs to be paired with the color in the recommended palette. In order to maintain the color distribution of the original image, we rearrange the order of the colors in the recommended palette to minimize the distance between the recommended palette and the original one. Let the object regions be the foreground and the rest of the image be the background, we transfer the color of the foreground from the original palette to the recommended one. In detail, we implement the color transfer using the method proposed in [5], which first transfers luminance based on the monotonicity concern, then transfers the chroma guided by the target palette. Here the color transfer algorithm is open to be replaced by other suitable methods. The transferred result is then blended with the background:

$$\Phi_i = \alpha_i * fg_i + (1 - \alpha_i) * bg_i. \tag{5}$$

Here α_i is the value of i-th pixel of the matte, which is linearly normalized to range $[0,1]$. fg_i is the color of the i-th pixel in the transferred foreground, while bg_i is the color of the i-th pixel in the background. As shown in the bottom-right of Fig. 2, the details of the results based on matting are obviously better than directly using the coarsely detected mask. It is worth noticing that our approach is also applicable for multi-objects in the image (see Fig. 5).

Global recoloring. As mentioned before, we use the pre-trained neural networks [6, 15] to detect and segment objects, while in some special cases the instances of an object cannot be effectively detected. For instance, the pre-trained network may find only some of the flowers in the flower sea. Our solution to this problem is a global recoloring strategy that changes all the regions that have the similar color as the detected objects (see Fig. 6).

More specifically, we define an extended palette with $k+x$ different colors for the whole image. First, k colors

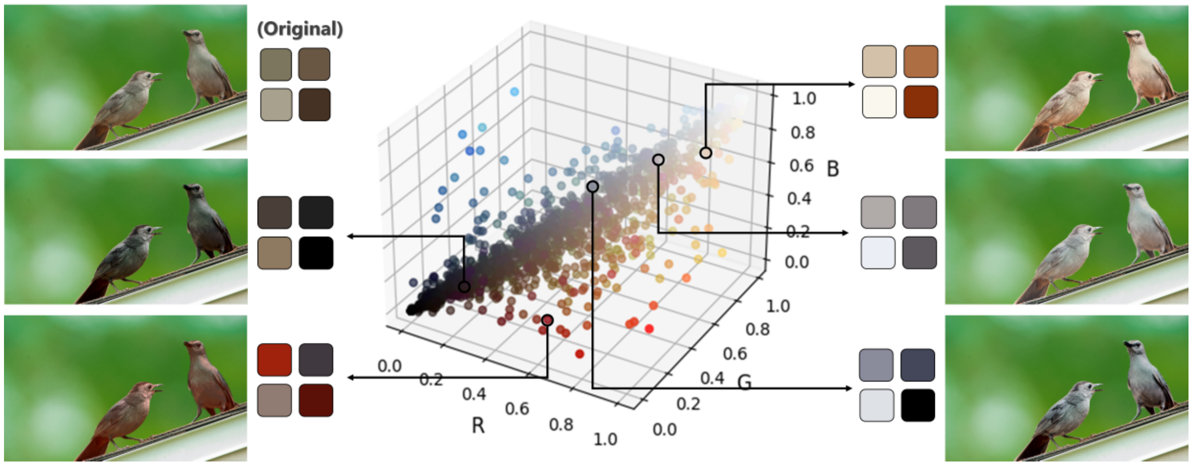

Fig. 7 Schematic diagram of our user interface. The interface provides a 3D diagram with color points of detected objects, in which a color point corresponds to a palette. When users select a point, the bird will be recolored directly using the corresponding recommended palettes. We also show some recoloring results on both sides of the picture.

are extracted from the masked regions of the image using our above-mentioned local recoloring method. The value of the number k is determined by the algorithm described in Sec. 3.1.1. Next, we select the minimum value of x whose corresponding *ratio* of the whole image is less than 0.20. After determine the value of k and x, we generate the extended palette. In the extended palette, k colors are recommended by the algorithm while other x colors are the same with the original palette. Finally, we transfer the color of the whole image with the extended palette.

User interface. We also design a user interface for more convenient operations of our instance-based recoloring approach. Given an input image, our approach first detects the objects, then the interface shows the 3D diagram of the corresponding palettes (see an example in Fig. 7). In the 3D diagram, a color point corresponds to a palette, and the color shown on the top has most adjacent colors to the reference objects. As shown in Fig. 7, our interface allows users to edit different natural colors of the input image interactively by just choosing the palette in the 3D diagram.

4 Experiments

4.1 Results

We use the train2017 images in COCO dataset to generate the candidate palettes for different object categories. With manually annotated masks for each image, we further filter out objects whose masks have less than 10000 pixels. After this step, 247820 palettes were generated for 80 categories. As mentioned in the previous section, for the online recoloring we use COCO dataset [30] trained Mask-RCNN [15] to obtain the mask of the input image. Some recoloring examples as well as the corresponding masks are shown in Fig. 8.

4.2 User study

In this part, we design subjective metrics to evaluate our approach. Since the recoloring task has no ground truth, it is not plausible to use common objective metrics (e.g. PSNR, SSIM) to evaluate our approach. To validate the effectiveness of our algorithms, we design a quantitative user study to compare the color editing results of our algorithm with another three algorithms [11, 43, 60].

4.2.1 Procedure and materials

First, 20 images with simple background are selected and processed by four algorithms. In detail, for each image and each algorithm, we generate 2 or 3 recoloring results, and 200 images are generated in total (see Fig. 10). The questionnaire is then designed to be two parts. In the first part, we examine whether the images are natural and artistic. 24 pictures are randomly selected (6 per algorithm) and shown for participants one by one. For each picture, the participants are asked to fill out a 7-point Likert scale with following questions:

- Please evaluate the realness of this picture (0 very unreal, 6 very real);
- Please evaluate the naturalness of this picture (0 very unnatural, 6 very natural);
- Whether this picture is visually pleasing (0 very unpleasing, 6 very pleasing);
- Whether the color of the picture is satisfactory (0 very unsatisfied, 6 very satisfied).

In the second part, we measure if the generated images can represent the possible colors of the objects. We use the original image and the generated 2 or 3 images with

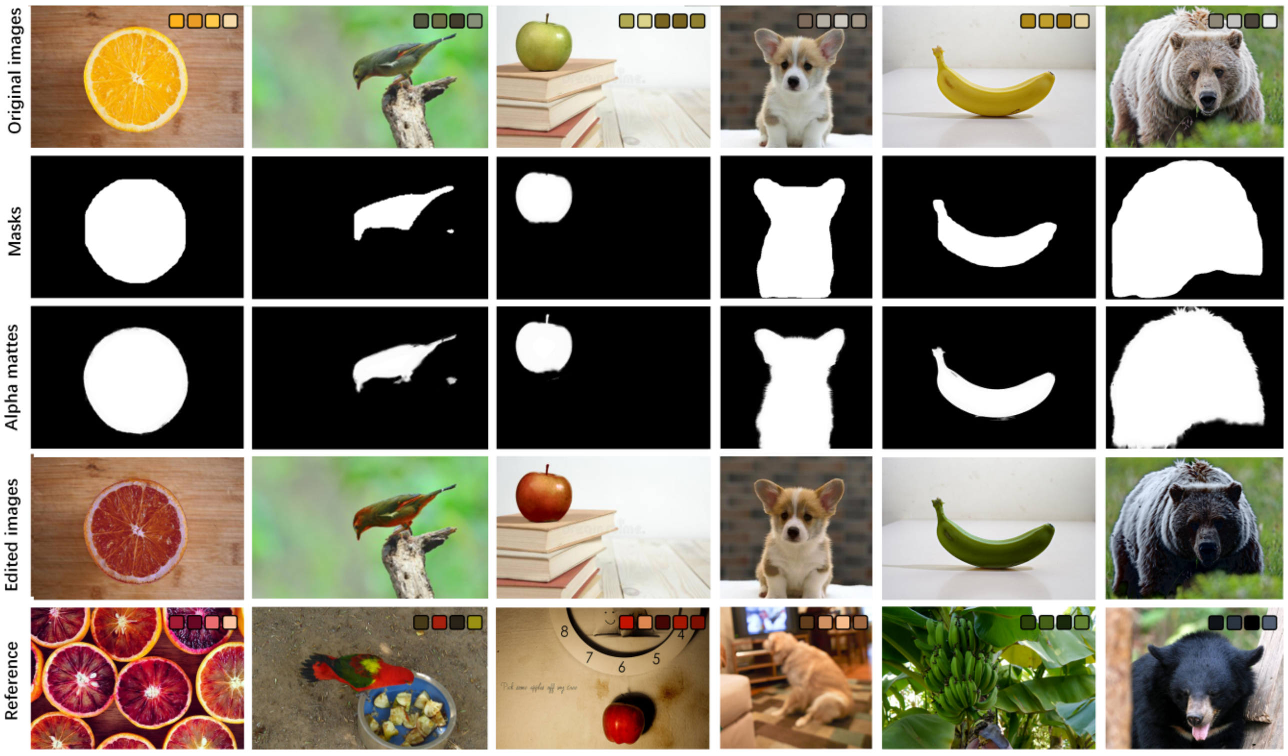

Fig. 8 Some results of our instance-based recoloring approach. The first row shows the original images with their original palettes at the top right. The second and third rows show the detected masks and their corresponding alpha mattes, respectively. The fourth row shows the color transferred results (see the respective target palettes and reference images in the last row).

one of the candidate algorithms (i.e. [60], [11], [43] and ours) to compose a testing set, and thus each image has four testing sets. For each participant, we randomly select an original image and its corresponding four testing set. The participants answer the question below for each testing set: *In the real world, the colors of the object have many possibilities; do you think this group of pictures can represent the possible colors of objects (0 not at all, 6 absolutely)?*

4.2.2 Participants and results

We published the questionnaires and received 82 answers, among which 80 are valid. The mean values of the scores are shown in Fig. 9. In general, our algorithm receives higher mean scores of all five questions.

Furthermore, we test if the difference is significant. For the four questions in the first part, we use the mean scores of six images which belong to the same algorithm and the same question as samples. The scores of samples for each question follow normal distribution tested by KolmogorovSmirnov test. Thus, we use student's t test to test the significance of the difference. As shown in Tab. 1, compared to other methods, the differences of our algorithm are statistically significant, which means that our method achieves higher realness, naturalness, and is more pleasing

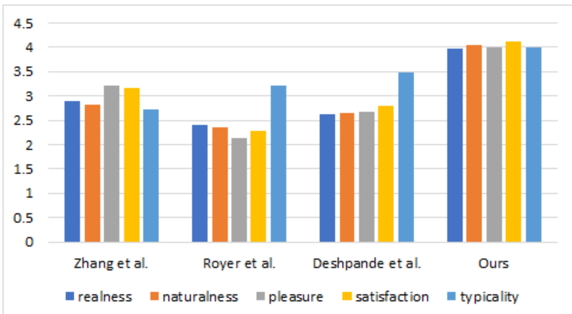

Fig. 9 Mean scores of each algorithm for five questions.

index	Zhang et al.	Royer et al.	Deshpande et al.
realness	4.409***	7.010***	5.869***
naturalness	4.966***	7.544***	6.039***
pleasure	3.354***	8.408***	5.922***
satisfaction	4.137***	8.358***	5.729***
typicality	17.881***	7.343***	3.668*

Tab. 1 The statistic of the difference between our method and others. In the form, $***$ means significance level 0.01, $**$ means significance level 0.05, $*$ means significance level 0.1.

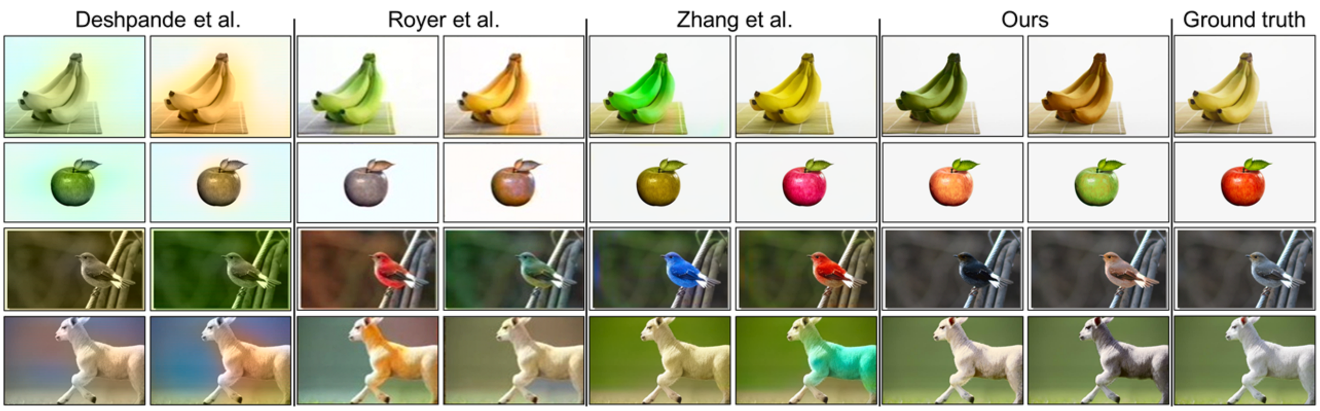

Fig. 10 The comparison of our method with Zhang *et al.* [60], Deshpande *et al.* [11], and Royer *et al.* [43]. Each row shows a series of colorization of one image, containing two colorization results for each method.

Fig. 11 Failure cases. Missing detection (the first row) or improper matting (the second row) may cause distortion.

for participants.

The scores of the question in the second part do not follow normal distribution, so we use Kruskal-Wallis H test to compare our method with others. As shown in Tab. 1, it proves that our method can produce more representative results for real-world objects.

5 Discussions

We have collected the palettes of different objects and proposed a natural recoloring approach. A two-part user study was performed to validate the effectiveness of our algorithm. In the first part, the analysis shows that our approach can produce more natural and realistic results while preserving the aesthetics. In the second part, we find that our method is more typical for the color modes of the natural objects in the real world. Our work can be easily applied in natural image editing with user interaction, and multiple recoloring options are also provided by the generated palettes. The global recoloring method can be further used in some color correction tasks.

Although our pipeline works well for most of natural objects, there are also some limitations of our work. Firstly, for some man-made objects the color distribution would be nearly arbitrarily, which makes the corresponding palette library less meaningful. Secondly, our pipeline is based on natural object detection, the missing or wrong detection will undermine the recoloring result. Some inaccurate matting results would also introduce undesirable edges of the object. In Fig. 11 we show two examples of the above-mentioned failure cases.

6 Conclusion

In this paper, we have extracted color patterns from the objects and created a palette set for different objects. A recoloring pipeline for natural color image editing was proposed to achieve more realistic and representative results. We also designed a user interface for convenient color editing. In the future, the scope of our work can be broadened to more kinds of objects with more complex color distributions, and the background (e.g. sky and grass) can also be taken into consideration.

Acknowledgements

This work was supported by NSFC (No. 61972216 and 62111530097) and Tianjin NSF (No. 18JCYBJC41300 and No. 18ZXZNGX00110). Shao-Ping Lu is the corresponding author of the paper.

References

[1] E. Aharoni-Mack, Y. Shambik, and D. Lischinski. Pigment-based recoloring of watercolor paintings. In *Proc. NPAR*, pages 1–11, 2017.

[2] X. An and F. Pellacini. Appprop: all-pairs appearance-space edit propagation. *ACM Trans. Graph.*, 27(3):1–9, 2008.

[3] X. An and F. Pellacini. User-controllable color transfer. *Comput. Graph. Forum.*, 29:263–271, 2010.

[4] H. Bahng, S. Yoo, W. Cho, D. K. Park, Z. Wu, X. Ma, and J. Choo. Coloring with words: Guiding image colorization through text-based palette generation. In *Proc. ECCV*, pages 431–447, 2018.

[5] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein. Palette-based photo recoloring. *ACM Trans. Graph.*, 34(4):139, 2015.

[6] K. Chen, J. Wang, J. Pang, Y. Cao, and et.al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[7] Q. Chen, D. Li, and C.-K. Tang. Knn matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(9):2175–2188, 2013.

[8] J. Cho, S. Yun, K. Mu Lee, and J. Young Choi. Palettenet: Image recolorization with given color palette. In *Proc. CVPR*, pages 62–70, 2017.

[9] Y.-Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proc. CVPR*, volume 2, page II, 2001.

[10] M.-Y. Cui, S.-P. Lu, M. Wang, Y.-L. Yang, Y.-K. Lai, and P. L. Rosin. 3D computational modeling and perceptual analysis of kinetic depth effects. *Computational Visual Media*, 6(3):265–277, 2020.

[11] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth. Learning diverse image colorization. In *Proc. CVPR*, pages 6837–6845, 2017.

[12] S. Duchêne, C. Aliaga, T. Pouli, and P. Pérez. Mixed illumination analysis in single image for interactive color grading. In *Proc. NPAR*, pages 1–10, 2017.

[13] H. S. Faridul, T. Pouli, C. Chamaret, J. Stauder, E. Reinhard, D. Kuzovkin, and A. Tremeau. Colour mapping: A review of recent methods, extensions and applications. *Comput. Graph. Forum.*, 35(1):59–88, 2016.

[14] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong. Image colorization using similar images. In *Proc. ACM MM*, page 369378, 2012.

[15] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *Proc. ICCV*, pages 2961–2969, 2017.

[16] Z. Hu, J. Jia, B. Liu, Y. Bu, and J. Fu. Aesthetic-aware image style transfer. In *Proc. ACM MM*, page 33203329, 2020.

[17] J. Huang, S. Zhou, X. Zhu, Y. Li, and C. Zhou. Automatic image style transfer using emotion-palette. In *International Conference on Digital Image Processing*, volume 10806, pages 1197 – 1206, 2018.

[18] Y. Huang, C. Wang, and C. Li. Translucent image recoloring through homography estimation. *Comput. Graph. Forum.*, 37(7):421–432, 2018.

[19] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.*, 35:1–11, 2016.

[20] R. Ironi, D. Cohen-Or, and D. Lischinski. Colorization by example. In *Rendering Techniques*, pages 201–210, 2005.

[21] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: A review. *IEEE Trans. Vis. Comput. Graph.*, 26(11):3365–3385, 2020.

[22] J. Kang and Y. Hwang. Hierarchical palette extraction based on local distinctiveness and cluster validation for image recoloring. In *Proc. ICIP*, pages 2252–2256, 2018.

[23] J. Lee, H. Son, G. Lee, J. Lee, S. Cho, and S. Lee. Deep color transfer using histogram analogy. *The Visual Computer*, 36(10):2129–2143, 2020.

[24] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graph.*, pages 689–694, 2004.

[25] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:228–42, 2008.

[26] B. Li, Y.-K. Lai, and P. L. Rosin. Sparse graph regularized mesh color edit propagation. *IEEE Trans. Image Process.*, 29:5408–5419, 2020.

[27] Z. Li, Z. Zha, and Y. Cao. Deep palette-based color decomposition for image recoloring with aesthetic suggestion. *MultiMedia Modeling*, pages 127–138, 2020.

[28] L. Liang, L. Jin, and X. Li. Facial skin beautification using region-aware mask. *IEEE trans. Systems, Man, and Cybernetics*, pages 2922–2926, 2013.

[29] S. Lin and P. Hanrahan. Modeling how people extract color themes from images. In *SIGCHI*, pages 3101–3110, 2013.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, and et.al. Microsoft COCO: Common objects in context. In *Proc. ECCV*, pages 740–755, 2014.

[31] H. Lu, Y. Dai, C. Shen, and S. Xu. Indices matter: Learning to index for deep image matting. In *Proc. ICCV*, pages 3266–3275, 2019.

[32] S. Lu, T. Mu, and S. Zhang. A survey on multiview video synthesis and editing. *Tsinghua Science and Technology*, 21(6):678–695, 2016.

[33] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens. Spatio-temporally consistent color and structure optimization for multiview video color correction. *IEEE Trans. Multimedia*, 5(17):577–590, 2015.

[34] S.-P. Lu, G. Dauphin, G. Lafruit, and A. Munteanu. Color retargeting: Interactive time-varying color image composition from time-lapse sequences. *Computational Visual Media*, 1:321–330, 2015.

[35] S.-P. Lu, S.-M. Li, R. Wang, G. Lafruit, M.-M. Cheng, and A. Munteanu. Low-rank constrained super-resolution for mixed-resolution multiview video. *IEEE Trans. Image Process.*, 30:1072–1085, 2021.

[36] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320, 2007.

[37] M. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research and Application*, 26(5):340–350, 2001.

[38] S. Ma, J. Fu, C. W. Chen, and T. Mei. Da-gan: Instance-level image translation by deep attention generative adversarial

networks. In *Proc. CVPR*, pages 5657–5666, 2018.
[39] R. Nguyen, B. Price, S. Cohen, and M. Brown. Group-theme recoloring for multi-image color consistency. *Comput. Graph. Forum.*, 36:83–92, 2017.
[40] P. O'Donovan, A. Agarwala, and A. Hertzmann. Color compatibility from large datasets. *ACM Trans. Graph.*, 30(4), 2011.
[41] H. Q. Phan, H. Fu, and A. B. Chan. Color orchestra: Ordering color palettes for interpolation and prediction. *IEEE Trans. Vis. Comput. Graph.*, 24(6):1942–1955, 2018.
[42] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.
[43] A. Royer, A. Kolesnikov, and C. H. Lampert. Probabilistic image colorization. *Proc. BMVC*, 2017.
[44] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Systems*, 42(3):19, 2017.
[45] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia. Deep automatic portrait matting. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proc. ECCV*, pages 92–107, 2016.
[46] J.-W. Su, H.-K. Chu, and J.-B. Huang. Instance-aware image colorization. In *Proc. CVPR*, pages 7968–7977, 2020.
[47] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *ACM Trans. Graph.*, volume 23, pages 315–321, 2004.
[48] X. Sun and J. Qin. Deep learning-based creative intention understanding and color suggestions for illustration. *Proc. ICAFE*, pages 90–96, 2020.
[49] J. Tan, S. DiVerdi, J. Lu, and Y. Gingold. Pigmento: Pigment-based image analysis and editing. *IEEE Trans. Vis. Comput. Graph.*, 25(9):2791–2803, 2019.
[50] J. Tan, J. Echevarria, and Y. Gingold. Efficient palette-based decomposition and recoloring of images via rgbxy-space geometry. *ACM Trans. Graph.*, 37(6):1–10, 2018.
[51] B. Wang, Y. Yu, T.-T. Wong, C. Chen, and Y.-Q. Xu. Data-driven image color theme enhancement. *ACM Trans. Graph.*, 29(6):1–10, 2010.
[52] Y. Wang, Y. Liu, and K. Xu. An improved geometric approach for palette-based image decomposition and recoloring. *Comput. Graph. Forum.*, 38(7):11–22, 2019.
[53] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. *ACM Trans. Graph.*, 21(3):277280, 2002.
[54] X. Wu, X.-N. Fang, T. Chen, and F.-L. Zhang. Jmnet: A joint matting network for automatic human matting. *Computational Visual Media*, 6(2):215–224, 2020.
[55] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu. Efficient affinity-based edit propagation using K-D tree. *ACM Trans. Graph.*, 28(5):1–6, 2009.
[56] L. Xu, Q. Yan, and J. Jia. A sparse control model for image and video editing. *ACM Trans. Graph.*, 32(6):1–10, 2013.
[57] N. Xu, B. Price, S. Cohen, and T. Huang. Deep image matting. In *Proc. CVPR*, pages 2970–2979, 2017.
[58] S. Ye, S.-P. Lu, and A. Munteanu. Color correction for large-baseline multiview video. *Signal Processing: Image Communication*, 53:40–50, 2017.
[59] Q. Zhang, C. Xiao, H. Sun, and F. Tang. Palette-based image recoloring using color decomposition optimization. *IEEE Trans. Image Process.*, 26(4):1952–1964, 2017.
[60] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph.*, 36(4):1–11, 2017.

Meng-Yao Cui is currently a Master student at Nankai University, majoring in Computer Science and Technology. Her research interests include visual perception and computing, human-computer interaction and machine learning.

Zhe Zhu is currently a Senior Research Associate at Duke University. He got his Ph.D. in the Department of Computer Science and Technology, Tsinghua University in 2017. He received his bachelors degree in Wuhan University in 2011. His research interests are in computer vision, computer graphics and medical image analysis.

Yulu Yang received the BE degree from Beijing Agriculture Engineering University, China in 1984. He received the ME and PhD degrees from Keio University, Japan in 1993 and 1996, respectively. He is currently a full professor in the Department of Computer Science, Nankai University, China. His research interests include parallel processing and intelligence computing.

Shao-Ping Lu is an associate professor of Computer Science at Nankai University in Tianjin, China. He had been a senior/postdoc researcher at Vrije Universiteit Brussel (VUB). He received his PhD degree in 2012 at Tsinghua University, China. Prior to that he also spent two years on high-performance SOC design in Shanghai. His research interests lie primarily in the intersection of visual computing, with particular focus on 3D video processing, visual scene analysis and visual forensics.