

Towards natural object-based image recoloring

Meng-Yao Cui¹, Zhe Zhu², Yulu Yang¹, and Shao-Ping Lu¹ (✉)

© The Author(s) 2021.

Abstract Existing color editing algorithms enable users to edit the colors in an image according to their own aesthetics. Unlike artists who have an accurate grasp of color, ordinary users are inexperienced in color selection and matching, and allowing non-professional users to edit colors arbitrarily may 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 distributions for different objects in the real world, and an online recoloring part that first recognizes the object category, and then recommends appropriate realistic candidate colors learned in the offline step for that category. We also provide an intuitive user interface for efficient color manipulation. After color selection, image matting is performed to ensure smoothness of the object boundary. Comprehensive evaluation on various color editing examples demonstrates 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 topic which has drawn widespread attention. By changing the color, we can change theme [1], style [2], illumination [3], and even emotional effect [4] 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 [5, 6] or statistics [7, 8]. 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 [9, 10], users only have to provide sparse inputs while the algorithm propagates the edits to appropriate regions in the rest of the image, based on pixel-level affinities. To further alleviate the user interaction burden, palette-based coloring [11] allows users to recolor an image by just editing a color palette. This is intuitive and users can adjust the color while given instant feedback. Previous palette-based recoloring methods mainly focus on the generation of the source palette [11–13] and the transfer from source colors to target colors [14]. They assume the target color is given or chosen according to user preference, which works for artists. For amateur users who do not have an accurate grasp of color, they cannot always select an appropriate color for a particular object, since they can distinguish between green and red, say, 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 color arbitrarily can lead to unrealistic editing results. One way to solve this problem is to use a reference picture [4] to generate the target palette. This approach is more efficient than directly choosing the color. However, finding a suitable reference picture is still a challenging task for users. There are also several colorization methods which provide diverse and realistic colorization results [15, 16], but their randomly generated results are insufficiently flexible for users. Our solution of this problem is based on the following observations: (i) The colors of objects in nature vary within a limited range, and the color of most objects can be represented by a few representative colors. For

1 TKLNDST, CS, Nankai University, Tianjin 300350, China. E-mail: M.-Y. Cui, cuimengyao@mail.nankai.edu.cn; Y. Yang, yangyl@nankai.edu.cn; S.-P. Lu, sl@nankai.edu.cn (✉).

2 Department of Radiology, Duke University, Durham, NC 27705, USA. E-mail: zhe.zhu@duke.edu.

Manuscript received: 2021-03-28; accepted: 2021-05-28

example, the color of a carrot is (normally) between yellow and red, while zebras have black and white stripes. (ii) Due to its large scale and diversity, in a large scale image dataset such as Microsoft COCO [17], the color distribution of an object category is already able to represent its actual color diversity 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 a large-scale image dataset in advance, and recommends 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 for apples in Fig. 1). Next, we merge the palettes for objects within the same object category and provide a default palette for each object category. Once the palettes have been generated, we can apply them to our online recoloring approach. For an input image, we detect the objects in the image, and generate alpha mattes with the detected masks. Next, we transfer the colors of the foreground using the recommended palettes and blend it with background. Our framework can also be applied to global color editing to meet other color editing requirements. Finally, a user interface is provided, in which users can edit the image intuitively and efficiently. Our main contributions are summarized as follows:

- a novel natural recoloring approach which recommends a target palette based on the color distribution of real objects, and
- 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 relevant work in those areas.

2.1 Palette-based color manipulation

Palette-based color manipulation edits the image by modifying the palettes. Such approaches can be divided into two groups. One group makes use of a pair of palettes for each image, the source palette and the target palette. One main way of generating the source palette is to cluster the colors in an image using the k -means method [11, 13, 18], which relies on the global color distribution. Kang and Hwang [8] provide an approach which can capture locally distinctive colors. However, methods based on RGB values are not adequate to recover all features in paintings. Thus, several works are designed to extract the original pigments in the paintings, which have different physical properties [12, 19]. Some methods further decompose the image into several mixing layers which correspond to the colors in the palette separately [5, 12]. Huang et al. [20] also provide a method which is designed for transparent object recoloring. Another group only needs a target palette, which is widely used in tasks like color compatibility [21], theme enhancement [22, 23], and recolorization [14]. This group of methods can also be used in colorization tasks. For instance, Bahng et al. [24] colorize a picture using a palette generated by text. Other applications such as transferring a color theme [1] and expressing different emotions [4] also implemented with palette-based methods.

2.2 Semantic-based color transfer

These color transfer methods recolor the image by taking one or more images as reference, which can be used for both colorization and recolorization. Many methods of color transfer are based on the statistics of color distributions [7, 8, 25], while there are also geometry-based methods [5, 6] and user-assisted solutions [26]. During color mapping, multi-

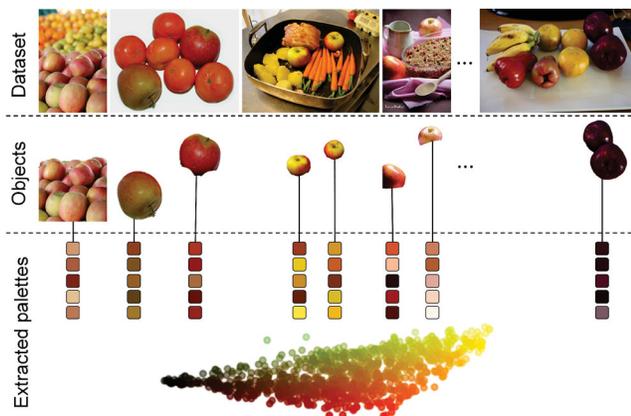


Fig. 1 Example of our palette extraction process. For the category “apple”, we first extract apples from images, and then calculate their palettes separately.

level semantic information is considered. Welsh et al. [27] match luminance and texture between images and transfer chromatic information. The method proposed in Ref. [28] 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 greater spatial consistency [29]. Furthermore, Iizuka et al. [30] combine global and local image features using a deep network, freeing the users from providing reference images. There are also several methods which make use of object-level information. The framework designed by Ma et al. [31] learns object-level correspondences for image translation. In colorization, colorizing instances and background separately and then merging the results with a fusion module improves the quality [32]. Facial skin beautification [33] also makes use of generated face masks.

2.3 Matting

Matting provides multiple utilities to separate the object from its background. Traditional matting methods try to solve the matting equation using a trimap or scribbles [34–37]. The results of these methods are highly dependent on color cues. In recent years, matting methods based on deep networks have been developed, using a convolutional neural network [38] which considers both semantic prediction and matte optimization. The dataset used for training is generated by a traditional matting method. Xu et al. [39] design a two-part model; the first part predicts mattes from an image and its trimap, and then they refine the alpha mattes in the second part. To solve human matting problems, Wu et al. [40] further propose an end-to-end learning framework which combines pose estimation with a trimap and matting network. Lu et al. [41] proposed a flexible network which can perform well in natural image matting tasks.

2.4 Color editing

Multiple tools have been designed for color editing. One class of color editing methods allows users to edit the color of all pixels in the image, using scribbles such as color points or strokes [42, 43] as input. Such methods were later improved in terms of speed [10, 44] and user effort [45]. Another kind of method implements higher-level image editing: for instance, some methods take a reference image to assist the

editing. In the framework provided by An and Pellacini [46], users draw pairs of strokes in target and reference images, to indicate regions with the same color “style”. The reference colors are transferred to the target image based on the stroke pair. Lu et al. [47] design a color retargeting approach, which composes a time-varying color image. Color editing has also been applied in multi-view [48–51] and 3D analysis [52, 53]. Some researchers also attempt to preserve aesthetic or creative intentions during transfer [54–56]. Palette-based recoloring is another popular approach to color editing, and as stated before, users can edit the image by modifying the colors in the palette. Most works focus on extraction of the original palette and the effect of recoloring, and think less about the selection of the target palette, with only a few methods recommending palettes for users. These recommendations can be based on the predicted color distribution [45] or online repositories [13]. Huang et al. [4] establish a dataset of palettes for different emotions. For each picture, palettes for different emotions can be recommended based on its original palette. Unlike these methods, our approach aims to provide natural and realistic recoloring with recommending palettes.

3 Approach

3.1 Methodology

Our approach consists of an offline candidate palette generation step and an online recoloring step. In the offline step we first generate thousands of initial palettes, and 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 categories of objects we provide candidate palettes for users. The pipeline of our approach is shown in Fig. 2. We use the images in the Microsoft COCO dataset [17] to generate the candidate palette for each object category in the offline step, since this is a large scale dataset, and it also provides masks for objects in 80 classes. Next, we give the details of those steps.

3.2 Masking

During online recoloring, we use Mask-RCNN [57] trained on the COCO dataset to extract masks of

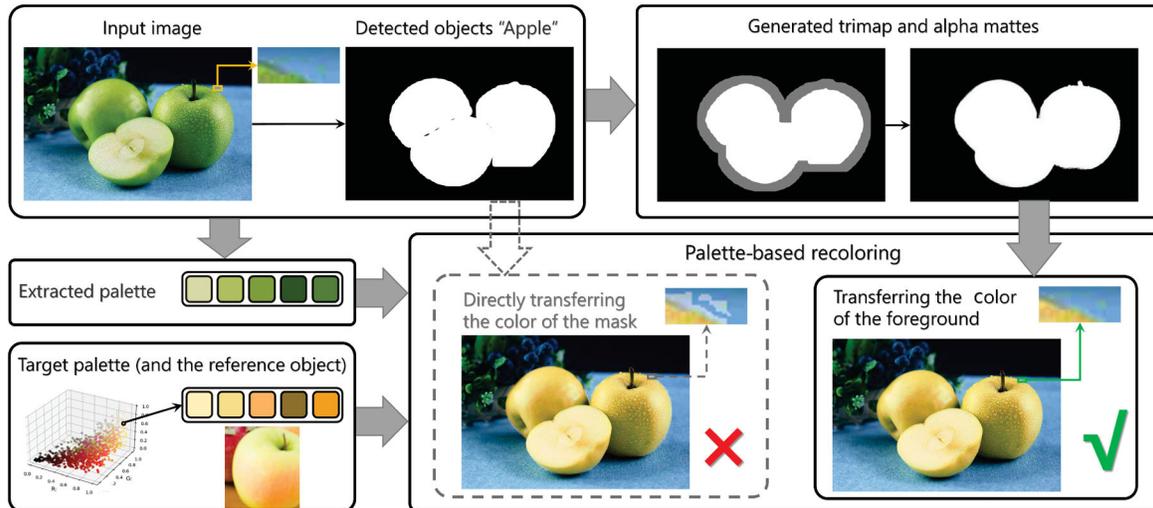


Fig. 2 Pipeline. We detect each object in the image and generate the palette based on the mask. We use alpha matting to process the mask and get the alpha matte of the object. Finally, we transfer the foreground color using the target palette chosen by the user.

the objects in the input image. The network is built with Open MMLab Detection [58]. As 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 operations 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 an unknown region, the corroded mask is regarded as the foreground region, and the rest of the picture is background. The trimap is then used to generate the alpha mattes using Indexnet [41]. This refinement is performed in both offline and online steps. An example of the improved segmentation results using alpha mattes is illustrated in Fig. 2.

3.3 Initial palette extraction

We use the method proposed by Chang et al. [11] to extract the initial palettes since it can be performed in real time. We extract the palettes for the 80 categories using the masks of the images of the dataset. More specifically, we first generate the palette for each masked region. In Ref. [11], the number of colors in the palette (denoted k in this paper) should be given beforehand. Instead of setting a fixed k for all palettes, we adopt an adaptive strategy by determining k for each category based on the following calculation. We first generate palettes with $k = 3, 4, 5, 6$ (larger k is slower) separately for all masked regions and calculate the loss of each palette

using the following equations:

$$d_i = \text{dist}(p_i, C_i) \quad (1)$$

$$\text{loss}_p = \|d\|_2^2$$

where p_i is the color of the i -th pixel of the masked region, C_i is a color from palettes corresponding to p_i , and $\text{dist}(\cdot, \cdot)$ calculates the color difference between two pixels using CIEDE2000, which is a closer metric to human assessment than RGB Euclidean distance [59]. Next, we calculate the ratio for each loss of the palettes as follows:

$$D_i = \text{dist}(p_i, C) \quad (2)$$

$$\text{ratio} = \text{loss}_p / \|D\|_2^2$$

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

3.4 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 the DBSCAN (density-based spatial clustering of applications with noise) method [60] to merge the

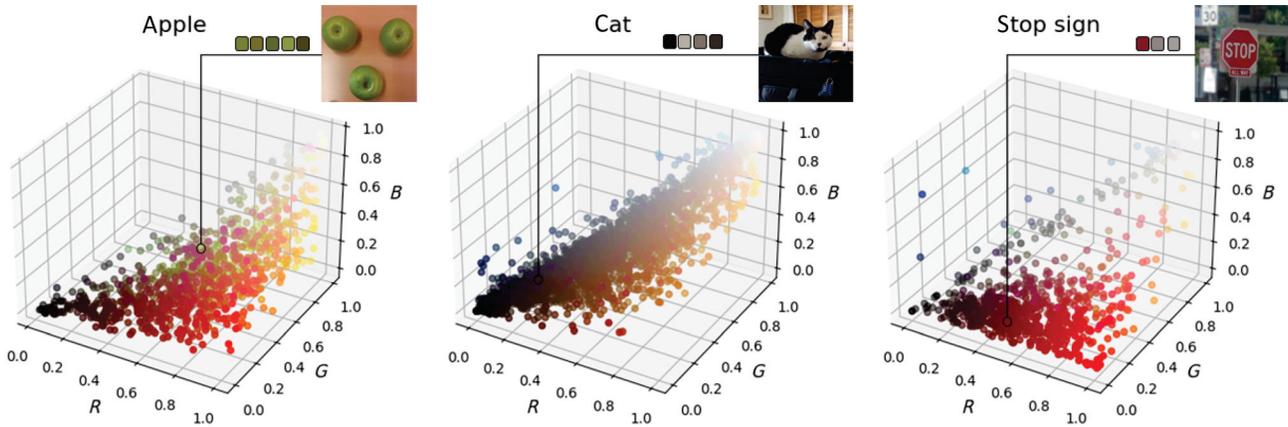


Fig. 3 Examples of generated palettes. For each palette, only one color with the most adjacent colors in the object is shown. The R, G, B values in the diagram are linearly normalized to the range $[0, 1]$. We also show the center palettes of each category with their corresponding images.

palettes (example results are shown 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 (S_d) comparing to 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:

$$\begin{aligned} d_j &= \text{dist}(x_j, r_{ij}) \\ \text{pd}_i &= \|d\|_2^2 \end{aligned} \tag{3}$$

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

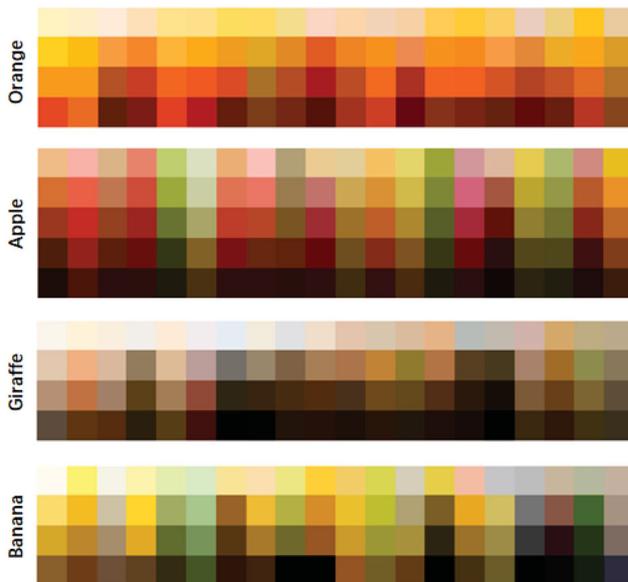


Fig. 4 Examples of recommended palettes. In each category, a column of colors corresponds to a recommend palette; 20 palettes are shown for each category.

distance between x and r_i . The colors in x and r_i are ranked by L values in LAB color space. S_d is then calculated using the following equation:

$$S_d = \sum_i^n \text{pd}_i \tag{4}$$

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).

3.5 Instance-based recoloring

In the online recoloring step, given an input image to recolor, we use the original palette extracted from the object and the recommended 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 color transfer using the method proposed in Ref. [11], which first transfers luminance based on monotonicity, and then transfers the chroma guided by the target palette. The color transfer algorithm could be replaced by other suitable methods. The transferred result is then blended with the background:

$$\Phi_i = \alpha_i f g_i + (1 - \alpha_i) b g_i \tag{5}$$

Here α_i is the value of the i -th pixel of the matte, linearly normalized to the 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 at the bottom-right of Fig. 2, results based on matting are obviously better in detail than those directly using the coarsely detected mask. Clearly, our approach is also applicable when there are multiple objects in the image (see Fig. 5).

3.6 Global recoloring

As mentioned before, we use pre-trained neural networks [57, 58] to detect and segment objects, but 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 a flowerbed. Our solution to this problem is to use a global recoloring strategy that changes all the regions with similar color to the detected objects (see Fig. 6).

More specifically, we define an extended palette



Fig. 5 Transfer results for images with multiple objects.

with $k + x$ different colors for the whole image. First, k colors are extracted from the masked regions of the image using our local recoloring method. The value of k is determined by the algorithm described in Section 3.3. Next, we select the minimum value of x whose corresponding *ratio* for the whole image is less than 0.20. After determining the values of k and x , we generate the extended palette. In the extended palette, k colors are recommended by the algorithm while x further colors are the same as in the original palette. Finally, we transfer the color of the whole image using the extended palette.

3.7 User interface

We have also designed a user interface to allow convenient operation of our instance-based recoloring approach. Given an input image, our approach first detects the objects, and then the interface shows a 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 at the

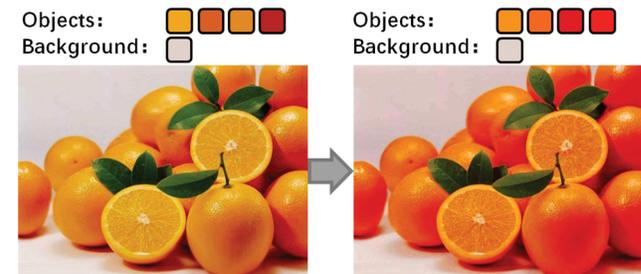


Fig. 6 Example for global recoloring. We extract five colors from the objects and a color from the background. The five colors of the objects are transferred while the colors of the background are preserved.

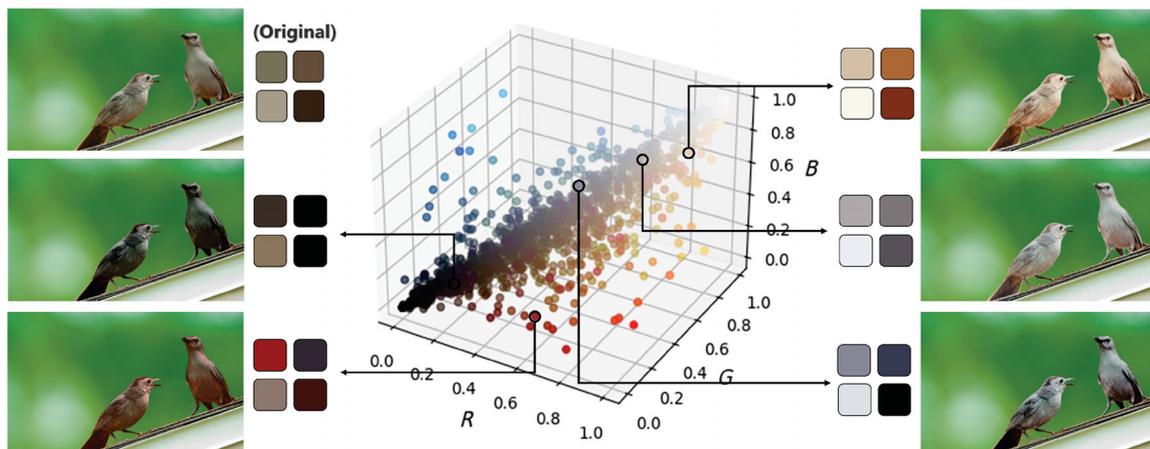


Fig. 7 Schematic diagram of our user interface. It provides a 3D diagram with color points for detected objects; a color point corresponds to a palette. When users select a point, the bird is recolored directly using the corresponding recommended palettes. We also show some recoloring results on both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

top has the nearest colors to the reference object. As shown in Fig. 7, our interface allows users to select and apply different natural colors to the input image interactively by just choosing the palette in the 3D diagram.

4 Experiments

4.1 Approach

We use the train2017 images in the 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 10,000 pixels. After this step, 247,820 palettes were generated for 80 categories. As noted earlier, for the online recoloring we use the COCO dataset [17] trained Mask-RCNN [57] to obtain masks for the input images. Some recoloring examples as well as corresponding masks are shown in Fig. 8.

4.2 User study

4.2.1 Procedure and materials

We designed 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 designed a quantitative user study to compare the color editing results of our algorithm with those from another three algorithms [15, 16, 45].

First, 20 images with simple backgrounds were selected and processed by four algorithms. In detail, for each image and each algorithm, we generated 2 or 3 recoloring results; 200 images were generated in total (see Fig. 9). The designed questionnaire had two parts. In the first, we examined whether the images are natural and artistic. 24 pictures were randomly selected (6 per algorithm) and shown to participants one by one. For each picture, the participants are asked to evaluate on a 7-point Likert scale:

- the realism of the picture (0 very unrealistic, 6 very realistic);
- the naturalness of the picture (0 very unnatural, 6 very natural);
- whether this picture is visually pleasing (0 very unpleasant, 6 very pleasant);
- whether the color of the picture is satisfactory (0 very unsatisfactory, 6 very satisfactory).

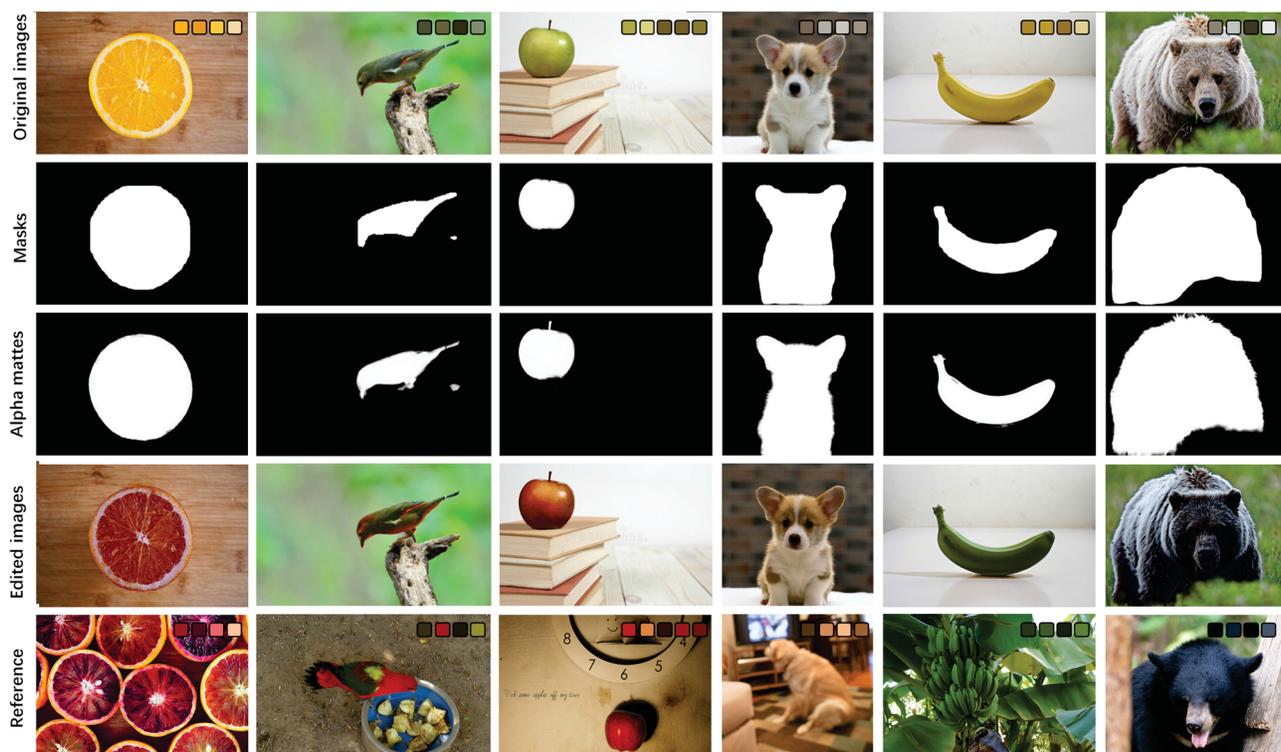


Fig. 8 Various results of our instance-based recoloring approach. Top row: original images with their original palettes at the top right. Rows 2, 3: detected masks and corresponding alpha mattes, respectively. Row 4: color transfer results. Row 5: target palettes used and corresponding reference images.

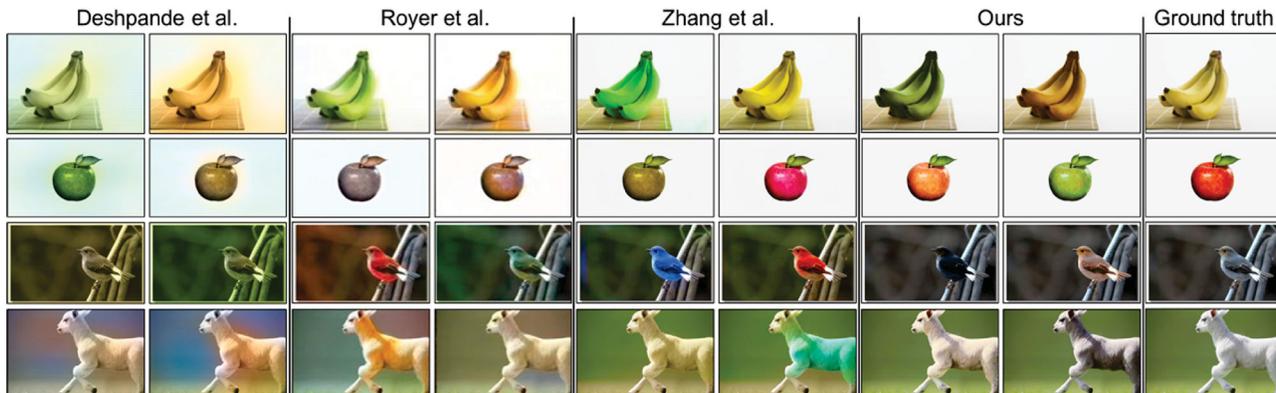


Fig. 9 Comparison of our method with those of Zhang et al. [45], Deshpande et al. [15], and Royer et al. [16]. Each row shows colorization of one image, giving two colorization results for each method.

In the second part, we measured whether the generated images can represent possible colors for the objects. We used the original image and the 2 or 3 images generated by one of the candidate algorithms to compose a test set, so each image has four test sets. For each participant, we randomly selected an original image and its corresponding four test sets. The participants answer the question below for each test set: *In the real world, the colors of an object have many possibilities; do you think this group of pictures can represent the possible colors of objects (0 not at all, 6 without doubt)?*

4.2.2 Participants and results

Our questionnaires received 82 responses, among which 80 were valid. The mean values of the scores are shown in Fig. 10. In general, our algorithm received higher mean scores for all five evaluations.

Furthermore, we may test if the differences are significant. For the four questions in the first part, we used the mean scores for six images which belong to the same algorithm and the same question as samples. The scores of samples for each question follow a normal distribution tested by the Kolmogorov–

Smirnov test. Thus, we use Student’s *t*-test to test the significance of the differences. As shown in Table 1, the differences between our algorithm and the others are statistically significant, so we conclude that our method achieves higher realism, naturalness, and more pleasing results for participants.

The scores of the question in the second part do not follow a normal distribution, so we use the Kruskal–Wallis *H* test to compare our method to others. Again, Table 1 shows that our method can produce more representative results for real-world objects.

Table 1 Statistics of the difference between our method and others. In the form, *** means significance level 0.01, ** means significance level 0.05, and * means significance level 0.1

Index	Zhang et al.	Royer et al.	Deshpande et al.
Realism	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*

5 Discussion

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 aesthetics. In the second part, we find that our method produces results more typical of the colors seen in natural objects in the real world. Our work can be easily used for interactive natural image editing, and multiple recoloring options are also provided by the generated

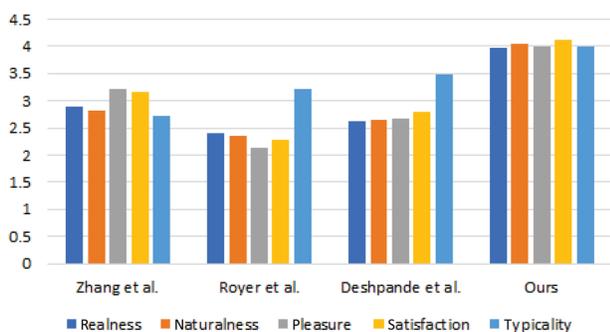


Fig. 10 Mean scores for each algorithm, for five evaluations.

palettes. The global recoloring method can be further used in some color correction tasks.

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



Fig. 11 Failures. Object detection failures (above) or improper matting (below) may cause unsatisfactory results.

6 Conclusions

In this paper, we have extracted color patterns from objects and created a palette set for different kinds of 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 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 National Natural Science Foundation of China (Grant Nos. 61972216 and 62111530097) and NSF of Tianjin City (Grant Nos. 18JCYBJC41300 and 18ZXZNGX00110).

References

- [1] Nguyen, R. M. H.; Price, B.; Cohen, S.; Brown, M. S. Group-theme recoloring for multi-image color consistency. *Computer Graphics Forum* Vol. 36, No. 7, 83–92, 2017.
- [2] Jing, Y. C.; Yang, Y. Z.; Feng, Z. L.; Ye, J. W.; Yu, Y. Z.; Song, M. L. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics* Vol. 26, No. 11, 3365–3385, 2020.
- [3] Duchêne, S.; Aliaga, C.; Pouli, T.; Pérez, P. Mixed illumination analysis in single image for interactive color grading. In: Proceedings of the Symposium on Non-Photorealistic Animation and Rendering, Article No. 10, 2017.
- [4] Huang, J.; Zhou, S. Z.; Zhu, X. Y.; Li, Y. W.; Zhou, C. F. Automatic image style transfer using emotion-palette. In: Proceedings of the 10th International Conference on Digital Image Processing, 1197–1206, 2018.
- [5] Tan, J. C.; Echevarria, J.; Gingold, Y. Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 262, 2019.
- [6] Wang, Y. L.; Liu, Y. F.; Xu, K. An improved geometric approach for palette-based image decomposition and recoloring. *Computer Graphics Forum* Vol. 38, No. 7, 11–22, 2019.
- [7] Lee, J.; Son, H.; Lee, G.; Lee, J.; Cho, S.; Lee, S. Deep color transfer using histogram analogy. *The Visual Computer* Vol. 36, Nos. 10–12, 2129–2143, 2020.
- [8] Kang, J. M.; Hwang, Y. Hierarchical palette extraction based on local distinctiveness and cluster validation for image recoloring. In: Proceedings of the 25th IEEE International Conference on Image Processing, 2252–2256, 2018.
- [9] An, X. B.; Pellacini, F. AppProp: All-pairs appearance-space edit propagation. *ACM Transactions on Graphics* Vol. 27, No. 3, 1–9, 2008.
- [10] Xu, K.; Li, Y.; Ju, T.; Hu, S. M.; Liu, T. Q. Efficient affinity-based edit propagation using K-D tree. *ACM Transactions on Graphics* Vol. 28, No. 5, 1–6, 2009.
- [11] Chang, H. W.; Fried, O.; Liu, Y. M.; DiVerdi, S.; Finkelstein, A. Palette-based photo recoloring. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 139, 2015.
- [12] Tan, J. C.; DiVerdi, S.; Lu, J. W.; Gingold, Y. Pigmento: Pigment-based image analysis and editing. *IEEE Transactions on Visualization and Computer Graphics* Vol. 25, No. 9, 2791–2803, 2019.

- [13] Zhang, Q.; Xiao, C. X.; Sun, H. Q.; Tang, F. Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing* Vol. 26, No. 4, 1952–1964, 2017.
- [14] Cho, J.; Yun, S.; Lee, K.; Choi, J. Y. PaletteNet: Image recolorization with given color palette. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 1058–1066, 2017.
- [15] Deshpande, A.; Lu, J. J.; Yeh, M. C.; Chong, M. J.; Forsyth, D. Learning diverse image colorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2877–2885, 2017.
- [16] Royer, A.; Kolesnikov, A.; Lampert, C. Probabilistic image colorization. In: Proceedings of the British Machine Vision Conference, 85.1–85.12, 2017.
- [17] Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision—ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740–755, 2014.
- [18] Phan, H. Q.; Fu, H. B.; Chan, A. B. Color orchestra: Ordering color palettes for interpolation and prediction. *IEEE Transactions on Visualization and Computer Graphics* Vol. 24, No. 6, 1942–1955, 2018.
- [19] Aharoni-Mack, E.; Shambik, Y.; Lischinski, D. Pigment-based recoloring of watercolor paintings. In: Proceedings of the Symposium on Non-Photorealistic Animation and Rendering, Article No. 1, 2017.
- [20] Huang, Y. F.; Wang, C. B.; Li, C. H. Translucent image recoloring through homography estimation. *Computer Graphics Forum* Vol. 37, No. 7, 421–432, 2018.
- [21] O'Donovan, P.; Agarwala, A.; Hertzmann, A. Color compatibility from large datasets. *ACM Transactions on Graphics* Vol. 30, No. 4, Article No. 63, 2011.
- [22] Lin, S.; Hanrahan, P. Modeling how people extract color themes from images. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 3101–3110, 2013.
- [23] Wang, B. Y.; Yu, Y. Z.; Wong, T. T.; Chen, C.; Xu, Y. Q. Data-driven image color theme enhancement. *ACM Transactions on Graphics* Vol. 29, No. 6, Article No. 146, 2010.
- [24] Bahng, H.; Yoo, S.; Cho, W.; Park, D. K.; Wu, Z. M.; Ma, X. J.; Choo, J. Coloring with words: Guiding image colorization through text-based palette generation. In: *Computer Vision—ECCV 2018. Lecture Notes in Computer Science, Vol. 11216*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 443–459, 2018.
- [25] Reinhard, E.; Adhikhmin, M.; Gooch, B.; Shirley, P. Color transfer between images. *IEEE Computer Graphics and Applications* Vol. 21, No. 5, 34–41, 2001.
- [26] Faridul, H. S.; Pouli, T.; Chamaret, C.; Stauder, J.; Reinhard, E.; Kuzovkin, D.; Tremeau, A. Colour mapping: A review of recent methods, extensions and applications. *Computer Graphics Forum* Vol. 35, No. 1, 59–88, 2016.
- [27] Welsh, T.; Ashikhmin, M.; Mueller, K. Transferring color to greyscale images. *ACM Transactions on Graphics* Vol. 21, No. 3, 277–280, 2002.
- [28] Irony, R.; Cohen-Or, D.; Lischinski, D. Colorization by example. In: Proceedings of the 16th Eurographics Conference on Rendering Techniques, 201–210, 2005.
- [29] Gupta, R. K.; Chia, A. Y. S.; Rajan, D.; Ng, E. S.; Huang, Z. Y. Image colorization using similar images. In: Proceedings of the 20th ACM International Conference on Multimedia, 369–378, 2012.
- [30] Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics* Vol. 35, No. 4, Article No. 110, 2016.
- [31] Ma, S.; Fu, J. L.; Chen, C. W.; Mei, T. DA-GAN: Instance-level image translation by deep attention generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5657–5666, 2018.
- [32] Su, J. W.; Chu, H. K.; Huang, J. B. Instance-aware image colorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7965–7974, 2020.
- [33] Liang, L. Y.; Jin, L. W. Facial skin beautification using region-aware mask. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2922–2926, 2013.
- [34] Chuang, Y. Y.; Curless, B.; Salesin, D. H.; Szeliski, R. A Bayesian approach to digital matting. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, II, 2001.
- [35] Chen, Q. F.; Li, D.; Tang, C. K. KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 35, No. 9, 2175–2188, 2013.

- [36] Levin, A.; Lischinski, D.; Weiss, Y. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 30, No. 2, 228–242, 2008.
- [37] Sun, J.; Jia, J.; Tang, C.-K.; Shum, H.-Y. Poisson matting. *ACM Transactions on Graphics* Vol. 23, No. 3, 315–321, 2004.
- [38] Shen, X. Y.; Tao, X.; Gao, H. Y.; Zhou, C.; Jia, J. Y. Deep automatic portrait matting. In: *Computer Vision—ECCV 2016. Lecture Notes in Computer Science, Vol. 9905*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 92–107, 2016.
- [39] Xu, N.; Price, B.; Cohen, S.; Huang, T. Deep image matting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2970–2979, 2017.
- [40] Wu, X.; Fang, X. N.; Chen, T.; Zhang, F. L. JMNet: A joint matting network for automatic human matting. *Computational Visual Media* Vol. 6, No. 2, 215–224, 2020.
- [41] Lu, H.; Dai, Y. T.; Shen, C. H.; Xu, S. C. Indices matter: Learning to index for deep image matting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3265–3274, 2019.
- [42] Levin, A.; Lischinski, D.; Weiss, Y. Colorization using optimization. *ACM Transactions on Graphics* Vol. 23, No. 3, 689–694, 2004.
- [43] Luan, Q.; Wen, F.; Cohen-Or, D.; Liang, L.; Xu, Y.-Q.; Shum, H.-Y. Natural image colorization. In: *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, 309–320, 2007.
- [44] Xu, L.; Yan, Q.; Jia, J. Y. A sparse control model for image and video editing. *ACM Transactions on Graphics* Vol. 32, No. 6, Article No. 197, 2013.
- [45] Zhang, R.; Zhu, J. Y.; Isola, P.; Geng, X. Y.; Lin, A. S.; Yu, T. H.; Efros, A. A. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 119, 2017.
- [46] An, X. B.; Pellacini, F. User-controllable color transfer. *Computer Graphics Forum* Vol. 29, No. 2, 263–271, 2010.
- [47] Lu, S. P.; Dauphin, G.; Lafruit, G.; Munteanu, A. Color retargeting: Interactive time-varying color image composition from time-lapse sequences. *Computational Visual Media* Vol. 1, No. 4, 321–330, 2015.
- [48] Lu, S. P.; Ceulemans, B.; Munteanu, A.; Schelkens, P. Spatio-temporally consistent color and structure optimization for multiview video color correction. *IEEE Transactions on Multimedia* Vol. 17, No. 5, 577–590, 2015.
- [49] Lu, S. P.; Mu, T. J.; Zhang, S. H. A survey on multiview video synthesis and editing. *Tsinghua Science and Technology* Vol. 21, No. 6, 678–695, 2016.
- [50] Ye, S. Q.; Lu, S. P.; Munteanu, A. Color correction for large-baseline multiview video. *Signal Processing: Image Communication* Vol. 53, 40–50, 2017.
- [51] Lu, S. P.; Li, S. M.; Wang, R.; Lafruit, G.; Cheng, M. M.; Munteanu, A. Low-rank constrained super-resolution for mixed-resolution multiview video. *IEEE Transactions on Image Processing* Vol. 30, 1072–1085, 2021.
- [52] Li, B.; Lai, Y. K.; Rosin, P. L. Sparse graph regularized mesh color edit propagation. *IEEE Transactions on Image Processing* Vol. 29, 5408–5419, 2020.
- [53] Cui, M. Y.; Lu, S. P.; Wang, M.; Yang, Y. L.; Lai, Y. K.; Rosin, P. L. 3D computational modeling and perceptual analysis of kinetic depth effects. *Computational Visual Media* Vol. 6, No. 3, 265–277, 2020.
- [54] Hu, Z. Y.; Jia, J.; Liu, B.; Bu, Y. H.; Fu, J. L. Aesthetic-aware image style transfer. In: *Proceedings of the 28th ACM International Conference*, 3320–3329, 2020.
- [55] Li, Z. Q.; Zha, Z. J.; Cao, Y. Deep palette-based color decomposition for image recoloring with aesthetic suggestion. In: *MultiMedia Modeling. Lecture Notes in Computer Science, Vol. 11961*. Ro, Y. et al. Eds. Springer Cham, 127–138, 2020.
- [56] Sun, X. H.; Qin, J. X. Deep learning-based creative intention understanding and color suggestions for illustration. In: *Advances in Artificial Intelligence, Software and Systems Engineering. Advances in Intelligent Systems and Computing, Vol. 1213*. Ahram, T. Eds. Springer Cham, 90–96, 2021.
- [57] He, K. M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988, 2017.
- [58] Chen, K.; Wang, J. Q.; Pang, J. M.; Cao, Y. H.; Xiong, Y.; Li, X. X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J. et al. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [59] Luo, M. R.; Cui, G.; Rigg, B. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research & Application* Vol. 26, No. 5, 340–350, 2001.
- [60] Schubert, E.; Sander, J.; Ester, M.; Kriegel, H. P.; Xu, X. W. DBSCAN revisited, revisited. *ACM Transactions on Database Systems* Vol. 42, No. 3, Article No. 19, 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. degree from the Department of Computer Science and Technology, Tsinghua University in 2017, and his bachelor degree from Wuhan University in 2011. His research interests are in computer vision, computer graphics, and

medical image analysis.



Yulu Yang received his B.E. degree from Beijing Agriculture Engineering University, in 1984. He received his M.E. and Ph.D. degrees from Keio University, Japan, in 1993 and 1996, respectively. He is currently a full professor in the Department of Computer Science, Nankai University. His research interests

include parallel processing and intelligent computing.



Shao-Ping Lu is an associate professor of computer science at Nankai University in Tianjin, China. He has been a senior/postdoc researcher at Vrije Universiteit Brussel (VUB). He received his Ph.D. 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 visual computing, with particular focus on 3D video processing, visual scene analysis, and visual forensics.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.