

RESEARCH ARTICLE

Color Feature Based Dominant Color Extraction

YOUNGHA CHANG¹, (Member, IEEE), AND NOBUHIKO MUKAI

Tokyo City University, Setagaya, Tokyo 158-8557, Japan

Corresponding author: Youngha Chang (ychang@tcu.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant JP22K12229.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethical Board of Tokyo City University.

ABSTRACT The dominant colors in an image can be used for image search, color editing, palette generation, and several other applications. Conventionally, dominant colors are extracted using clustering or histogram-based methods. However, these methods cannot extract the dominant colors of small regions, which are essential for the analysis of color schemes. This study developed an approach to automatically extract dominant colors based on color features that are typically considered by human observers when analyzing color schemes. The proposed method first calculates the initial dominant color candidates using the K-means algorithm in the CIELAB color space and the graph cut of a region adjacency graph (RAG) of the segmented image. Next, the algorithm calculates the color features such as the saturation, contrast, and area of each cluster, based on which it extracts the dominant colors. Our method can extract prominent colors from small image regions as the dominant colors, which is not possible using conventional methods.

INDEX TERMS Color, image color analysis, image processing.

I. INTRODUCTION

Color plays a critical role in images; even a limited set of colors can represent an entire image. This “set” of dominant colors is called a color palette and is central to various research tasks such as retrieving images using dominant colors [1], [2], [3] or color sketches [4], analyzing image color strategies [5], [6] or image color harmony [7], and transferring colors using dominant colors [8], [9]. Most of the above-mentioned studies use the histogram or clustering-based methods [10]. These methods tend to prioritize the colors that occupy a large image area. However, when analyzing the color scheme of an image, colors that stand out despite occupying a small area must be considered. For example, Fig. 1 shows an image that is almost completely black or gray, with a small ochre-colored moon. When analyzing this image, histogram-based or clustering-based methods select only grays with various brightness levels as the dominant colors. However, the contrast between the black background and ochre region in this image is essential.

This study proposes a new dominant color extraction method to solve this problem. In this method, we first analyze

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar¹.

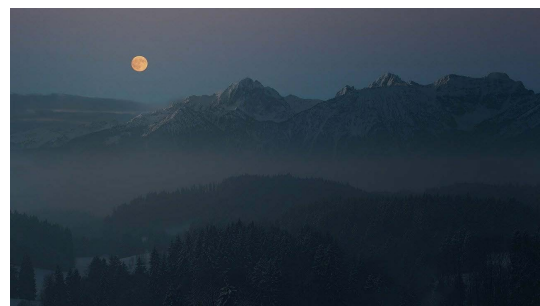


FIGURE 1. Example of an image with a prominent color occupying a small area.

the features of existing human-made color palettes. Based on the results of the analysis, our algorithm selects the final dominant colors.

II. PREVIOUS WORKS

Color schemes are a staple of research in the field of design. Numerous books [11], [12], [13] and online services [14], [15] provide examples of color schemes in images, designs, and paintings. However, these sources provide only short descriptions of terms such as “base colors” and “accent colors”. Quantitative descriptions of how to extract dominant colors are not available.

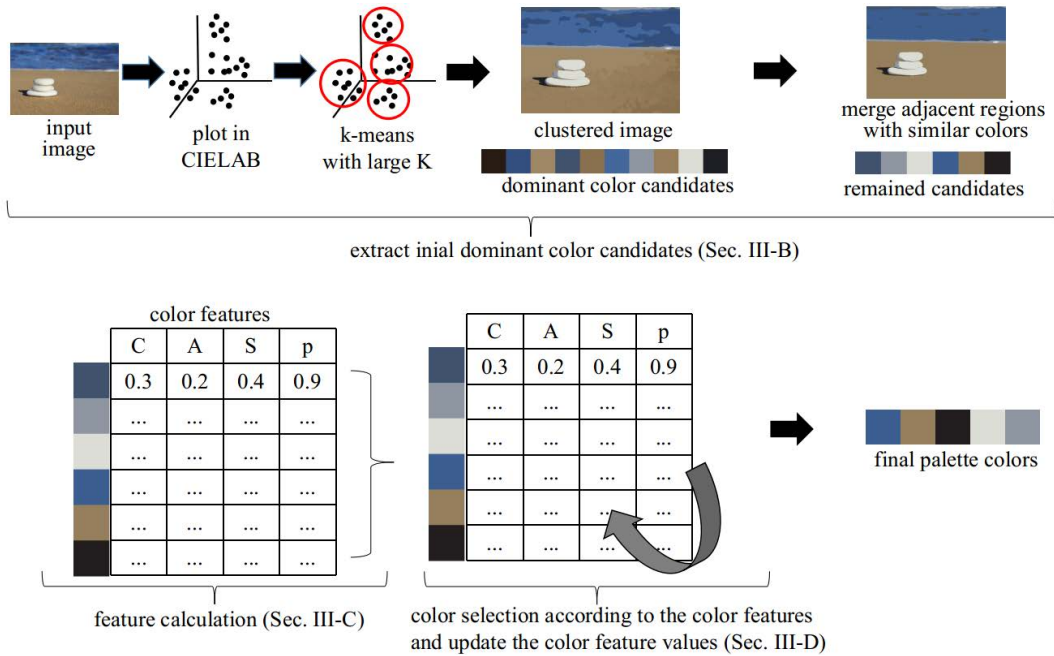


FIGURE 2. Overview of the proposed algorithm.

Several methods for dominant color extraction from images have been reported in previous studies. For example, the dominant color descriptor (DCD), one of the descriptors in MPEG-7, describes representative colors in an image [16]. This method first quantizes image colors using the generalized Lloyd algorithm (GLA) and selects a maximum of 8 colors as the dominant colors. However, it sometimes fails to retrieve conspicuous colors that are restricted to a small region within the image. Similarly, Chang *et al.* proposed a method to extract color palettes and use them to transfer image colors [8]. They used the K-means algorithm to extract dominant colors. In addition, they extended the K-means to obtain a stable initial cluster center with various colors. However, this method occasionally fails to select conspicuous colors limited to small regions. Furthermore, Chang *et al.* proposed a method [17] to extract a compact set of dominant colors using the K-means algorithm and the basic color categories [18]. In this method, a large number of initial candidate colors are extracted. Subsequently, the candidate colors are merged until there are no more colors having only a small color difference or belonging to the same basic color categories. Although this method can produce a compact set of dominant colors, the number of dominant colors tends to be smaller than expected. Other studies reported methods that decompose the colors of the image into multiple RGBA layers for color manipulation. Each palette color corresponds to a single color layer. Therefore, the colors of the given image can be manipulated by changing the palette colors. The easiest method to obtain the palette color is the K-means algorithm in the RGB color space [19]. Certain studies use a convex hull that encloses all colors in the image and regard the vertices of the convex hull as the candidate palette colors. Subsequently, the final palette colors are acquired

by simplifying the convex hull [20]. In convex hull-based approaches, colors tend to be saturated.

III. PROPOSED METHOD

In this section, a dominant color extraction method is explained.

A. BASIC STRATEGIES

To devise an effective dominant color extraction method, we analyzed more than 500 data sets of color palettes from images extracted manually. Based on this analysis, the following hypotheses are developed:

- 1) The number of dominant colors in most cases ranges from 3 to 6.
- 2) Colors with high saturation are often selected. In particular, the most vivid color is almost always selected.
- 3) The color occupying the largest area is almost always selected regardless of its conspicuousness.
- 4) If several colors stand out to some extent, colors with different hues are more likely to be selected.
- 5) The color that is significantly different from its surroundings is more likely to be selected.

To obtain an algorithm that extracts the dominant colors based on the above features, the following steps must be followed, resulting in the extraction of a 5-color palette.

- 1) Extract a greater number of candidate colors than the final ones (Sec. III-B).
- 2) Calculate the saturation, area, and color contrast to the surroundings of each candidate color (Sec. III-C).
- 3) Select the final dominant colors from the candidates according to their color features (Sec. III-D).

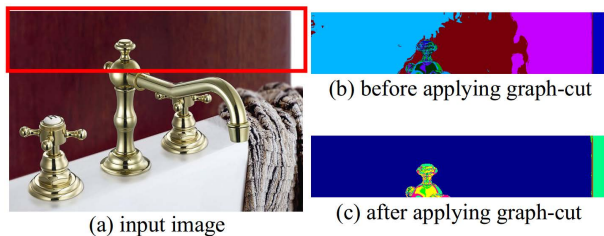


FIGURE 3. Before and after consideration of the image space coherency.

Fig. 2 shows the overview of the algorithm. The algorithm first performs the clustering in the CIELAB color space and merges adjacent image regions having similar colors. The cluster center colors become the initial dominant color candidates. Subsequently, the algorithm calculates the color features for each dominant color candidate. Finally, the final dominant colors are obtained based on the calculated color features.

B. EXTRACTION OF INITIAL DOMINANT COLOR CANDIDATES

The algorithm first applies the bilateral filter [21] to the image for noise reduction. The filtered pixel value $g(i, j)$ is calculated as shown in (1).

$$g(i, j) = \frac{\sum_{n=-w}^w \sum_{m=-w}^w f(i+m, j+n)e(i, j, m, n)}{\sum_{n=-w}^w \sum_{m=-w}^w e(i, j, m, n)},$$

$$e(i, j, m, n) = \exp\left(-\frac{m^2 + n^2}{2\sigma_1^2} - \frac{(f(i, j) - f(i+m, j+n))^2}{2\sigma_2^2}\right),$$
(1)

where $f(i, j)$ is the pixel value of (i, j) in the CIELAB color space, w is the kernel size, σ_1 and σ_2 are the smoothing parameters, and $e(i, j, m, n)$ approaches 0 when the color difference is large or m and n become large. Next, we apply the K-means algorithm. The initial cluster centers are determined by [8], but the darkest cluster is not deleted. The K-means algorithm considers coherency in the color space and ignores it in the image space. Fig. 3 shows an example. Fig. 3(a) is an input image. Here, we focus on the brown wall enclosed by the red square. It seems to have similar colors, but if we apply the K-means to the image, the wall is segmented into 3 parts as shown in Fig. 3(b). In Fig. 3(b), the different colors indicate different color clusters.

To consider coherency in the image space, we generate a region adjacency graph (RAG) of the segmented image and apply the graph cut method to merge similar adjacent regions. Fig. 3(c) shows the result after the graph cut segmentation. The three separated wall areas, cyan, brown, and magenta in Fig. 3(b), are combined into one region.

Fig. 4(a), (b), and (c) show an input image, the result of the K-means and the RAG graph cut, and the acquired initial dominant color candidates that are cluster center colors, respectively. In Fig. 4(b), each pixel is painted with its

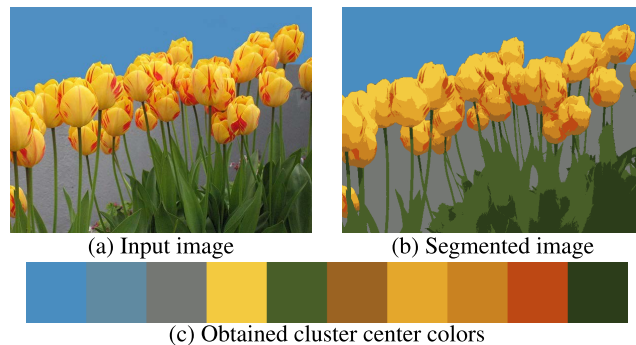


FIGURE 4. Initial dominant color candidates.

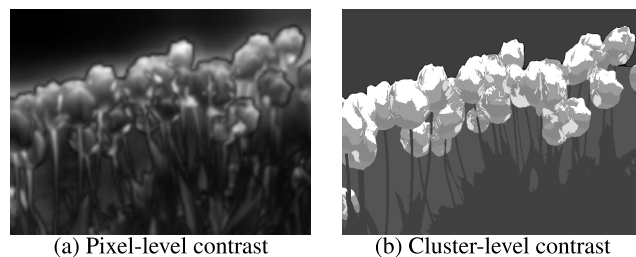


FIGURE 5. Pixel-level and cluster-level color contrasts.

associated cluster color that is one of the initial candidate colors. Hereafter, the term ‘‘candidate color’’ k is used synonymously with ‘‘cluster’’ k .

C. CALCULATION OF COLOR FEATURES

In this section, we calculate the color contrast, area occupied in the image, and saturation of each candidate color. The color contrast is calculated using Itti’s attention map [22]. In this method, the hue, luminance, and edge direction contrasts are calculated at various image scales to obtain both local and global image contrasts.

The contrast C_k of each candidate color k is defined in (2).

$$C_k = \frac{1}{N_k} \sum_{(i,j) \in k} C(i, j),$$
(2)

where N_k is the number of pixels belonging to the cluster k , and $C(i, j)$ ($0 \leq C(i, j) \leq 1$) is Itti’s attention value for pixel (i, j) . Fig. 5(a) shows an example of Itti’s original method applied to Fig. 4(a). The brighter pixels have a higher contrast. Fig. 5(b) shows each cluster contrast by painting pixel (i, j) according to the value of C_k (pixel $(i, j) \in k$).

Following the color contrast, the area occupied by each dominant color candidate is calculated by counting the number of pixels belonging to cluster k . The area A_k occupied by candidate color k is defined as shown in (3).

$$A_k = \text{the number of pixels belonging to cluster } k$$
(3)

Then each A_k is normalized using the maximum value of A for all candidate colors.

In addition, the saturation S_k of a candidate color k is defined as shown in (4).

$$S_k = \sqrt{a_k^2 + b_k^2},$$
(4)

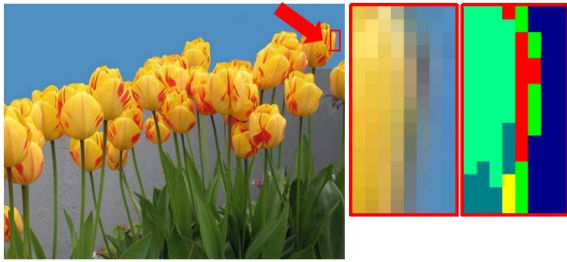


FIGURE 6. Color mixture near the boundaries.

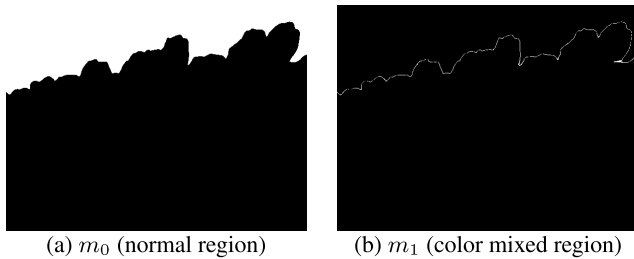


FIGURE 7. Examples of the mask image.

where a_k and b_k are a^* and b^* values in CIELAB of a dominant color candidate k , respectively. Then each S_k is normalized using the maximum value of S for all candidate colors.

The integrative color feature p_k is computed as shown in (5).

$$p_k = C_k + A_k + S_k \tag{5}$$

D. DETERMINATION OF THE FINAL DOMINANT COLORS

In this section, we extract the final dominant colors according to the color features.

When the background and object colors differ considerably, the boundary often forms an independent cluster after the K-means algorithm and RAG processing. Therefore, the boundary colors are eliminated prior to the dominant color extraction. Fig. 6 shows an example.

The image on the left is the input image, which is partially magnified and presented in the middle. In the middle image, a color mixture exists between the yellow flower and the blue background. The right shows the clustered image. Different colors indicate different color clusters. The mixing of colors creates a new cluster, i.e., the area shown in red. The color contrast of this cluster is high. Thus, there is a high possibility that it will be extracted as the final dominant color. Therefore, the boundary colors should be deleted prior to the extraction. In general, boundary clusters form extremely thin image regions. Therefore, these boundary clusters can be deleted by applying an erosion operation to each cluster region. For this purpose, we first generate a mask image for each cluster using (6).

$$m_k(i, j) = \begin{cases} 1 & \text{if pixel } (i, j) \in k, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Fig. 7 shows two examples of mask images, m_0 and m_1 , for Fig. 4(a). Black represents 0, and white represents 1.



FIGURE 8. Eliminate a candidate color caused by color mixing.



FIGURE 9. Final dominant colors for Fig. 4(a).

The mask image m_0 is the image in which the pixels belonging to cluster 0 corresponding to the blue sky in Fig. 4(a) are painted in white, whereas the mask image m_1 shows the pixels belonging to cluster 1 corresponding to the boundary between the yellow flower and the blue sky are painted in white. Cluster 1 is generated by mixing the yellow flower and blue sky. It appears between the color boundaries and are very narrow. Erosion is then applied to all the mask images m_k . If the white pixels disappear after the erosion, it seems that the candidate color k appears due to color mixture, and our algorithm deletes it from the candidate color list.

Fig. 8 shows an example. In Figs. 4(a) and (c), cluster 1 is detected as a color mixing cluster, and we delete the corresponding candidate color from the candidate list.

Next, the final dominant colors are selected based on the findings presented in Section III-A. The final dominant colors are determined based on saturation, contrast, and area size:

- 1) Select the color k with the highest saturation S_k .
- 2) Select the color k with the largest area A_k .
- 3) Repeatedly select colors k based on the p_k values until 5 colors are selected

Whenever we choose a candidate color as the final dominant color d , we multiply the weights $w(d, k)$ to all feature values of candidate colors k , i.e., S_k, A_k, C_k and p_k . $w(d, k)$ is defined as shown in (7).

$$w(d, k) = w_1(d, k)w_2(d, k), \tag{7}$$

$$w_1(d, k) = 1 - \exp^{-\Delta_h^2/\sigma_h^2}, \tag{8}$$

$$w_2(d, k) = 1 - \exp^{-\Delta_c^2/\sigma_c^2}, \tag{9}$$

where, Δ_h is the hue angle difference between d and k . σ_h is the standard deviation of Δ_h for all candidate colors. Δ_c is the CIELAB color difference between d and k , and σ_c is the standard deviation of Δ_c for all candidate colors.

$w_1(d, k)$ and $w_2(d, k)$ become approximately 1 if the hue difference and color difference between d and k are large, respectively. Therefore, we can avoid selecting similar colors and colors having hues similar to d .

Fig. 9 shows the final dominant colors for Fig. 4(a). The leftmost yellow color is extracted because its saturation is the highest among the candidate colors (Fig. 8). The second blue color is extracted because its area is the largest. The remaining colors are extracted based on the p_k values.

IV. EXPERIMENTAL RESULTS

We downloaded 300 images from the Internet and extracted the dominant colors using our method alongside the methods reported in previous studies. Fig. 10 shows the results. Each

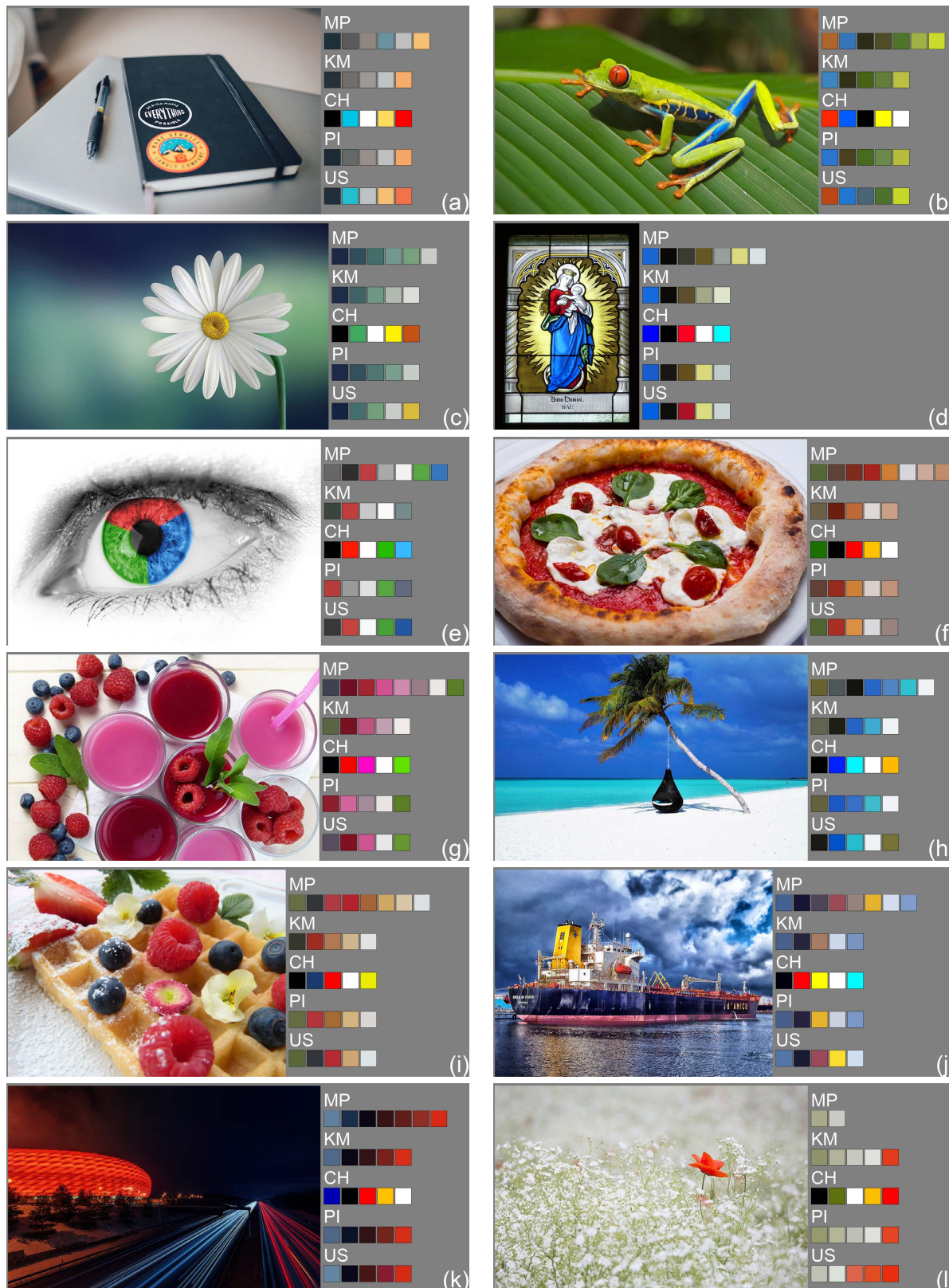


FIGURE 10. Extracted color palettes. (a)-(d): results for images having small but prominent color regions, (e)-(j): results for colorful images, (k): an example having dominant thin color regions, (l): an example where our method is inferior to other methods.

dominant color arrangement was sorted using the method described in [23]. In each image, the image on the left is

an input image and the color sets on the right are the dominant colors extracted by the MPEG-7 DCD [24] (MP), RGB

K-means algorithm used in [19] (KM), a convex hull based approach [20] (CH), a method described in [8] and [25], (PI), and our method (US), respectively.

Fig. 10(a)-(d) demonstrate that our method can extract prominent colors that occupy only a small area. In Fig. 10(a), the mark on the notebook is prominent. It comprises a thin cluster of deep oranges and a small region of turquoise blue. As expected, US succeeds in extracting them. In Fig. 10(b), the contrast between red-eye and green leaves is dominant. As expected, US can capture this feature. In Fig. 10(c), the small yellow region is important for analyzing the color scheme. Again, US can capture this. In Fig. 10(d), yellow, blue, and red are significant. US can capture all of them. In contrast, the conventional methods fail to extract a few of the dominant colors in the input images. Consequently, we can infer that the US can capture dominant colors that occupy only a small area.

Fig. 10(e)-(j) show the results for colorful images. For example, Fig. 10(e) possesses a significant color contrast between black and white, or chromatic and achromatic colors. US can extract the saturated red, green, and blue colors from this image. In Fig. 10(f), the green leaves, the white cheese, and the red tomato sauce form color harmony. Our method captures these colors. Fig. 10(g)-(j) also show that our method can capture hue variations, including prominent achromatic colors such as white and black.

Fig. 10(k) shows an example having dominant thin color regions. In US, erosion is applied to ignore color mixing at the region boundaries. However, we can successfully obtain the color occupying the narrow region because the number of erosions is small.

Fig. 10(l) shows an example where US is inferior to the other methods. A more achromatic color gradation is expected in the color palette of this image rather than in the color red. The S_k and C_k values of red flowers are very high. Consequently, the p_k of reddish colors are considerably higher than those of achromatic colors. Therefore, reddish colors are more likely to be extracted than whitish colors.

Similarly, if most of the image is covered with achromatic colors with only a small saturated color area as one point, people might select one dominant color from the one-point color and then select the remaining dominant colors from the achromatic colors. However, in the proposed method, multiple colors are often selected from the saturated colors.

Following the comparative study, we conducted subject experiments to investigate the effectiveness of the proposed method.

All methods except MPEG-7 DCD extract 5 colors. MPEG-7 DCD extracted 2 to 8 dominant colors from 300 images. To ensure consistency, we used only those images from which a similar number of representative colors were extracted. Using MPEG-7 DCD, 4 colors were captured in 5 images, 5 colors were captured in 11 images, and 6 colors were captured in 30 images. Then, we used these 46 images for the testing.

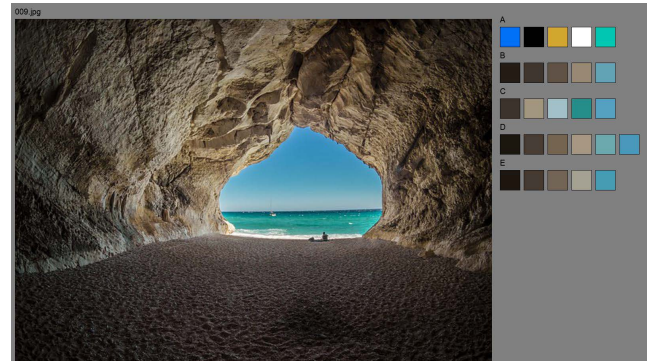


FIGURE 11. Monitor image in the experiment.

TABLE 1. Experimental results.

	MP	KM	CV	PI	US
Total	541	511	363	551	557

In the experiment, an EIZO ColorEdge CG279X, a 27 inch sRGB calibrated monitor, was used. The subjects positioned themselves in front of the monitor at a distance of 100 cm. In the monitor, the left side displayed the image, and the right side displayed five types of extracted color palettes as shown in Fig. 11. The order of the color palettes was shuffled for each image.

The directions provided to the subjects are as follows.

<Direction>

We want to analyze the color scheme of the images based on the dominant colors present in them. Five different color palettes are shown on the monitor. Please score each palette on a scale from 1 to 5. Each score indicates the following:

- 5 : Perfect or needs slight modification.
- 4 : Fairly suitable.
- 3 : Moderate.
- 2 : Needs significant modification.
- 1 : Not usable at all.

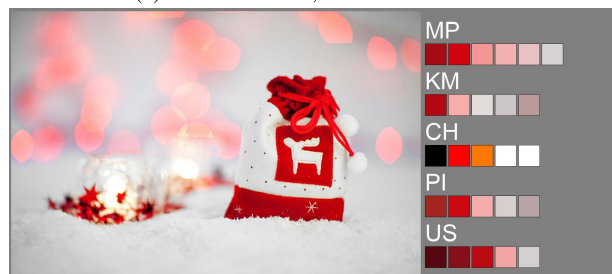
Six Japanese participants in their 20s participated in this experiment. The Ishihara test plate was used to confirm that all participants have a normal color vision [26]. Each participant took approximately 20 min. to assess 23 images, i.e., half of the entire dataset (46 images). In other words, each image was tested by 3 participants.

Table 1 shows the total scores obtained using each method. These scores show that PI and US produced the best results.

Fig. 12(a) shows the case where the US score was better than that of PI. As shown in this example, US (the proposed method) extracts small areas that include prominent colors. On the other hand, Fig. 12(b) shows a case where PI performs better than US. The lower US score is attributed to the same reason as described in Fig. 10(l). This image comprises large achromatic regions with small red regions. Consequently, S_k and C_k become large in the reddish regions. Therefore, our algorithm produces multiple reddish dominant colors. This problem should be addressed in future research.



(a) PI total score:7, US total score:12



(b) PI total score:12, US total score:9

FIGURE 12. Images with different scores between PI and US.

V. CONCLUSION

In this study, we proposed a method to extract dominant colors based on the color contrast, area, and saturation of images. In contrast to previous methods, our method was able to extract prominent colors from small regions, which is crucial in color scheme analysis. However, the proposed method is limited to identifying the dominant colors by simply summing the color features, as shown in (5). In the future, we plan to develop a method that can more accurately assess image regions by reconsidering how to combine color features. Furthermore, we plan to apply our method to achieve an automatic color scheme analysis.

REFERENCES

[1] A. Talib, M. Mahmuddin, H. Husni, and L. E. George, "A weighted dominant color descriptor for content-based image retrieval," *J. Vis. Commun. Image Represent.*, vol. 24, no. 3, pp. 345–360, Apr. 2013.

[2] TinEye. *Multicolor*. Accessed: Mar. 10, 2022. [Online]. Available: <http://labs.tineye.com/multicolor/>

[3] G. Xie, B. Guo, Z. Huang, Y. Zheng, and Y. Yan, "Combination of dominant color descriptor and Hu moments in consistent zone for content based image retrieval," *IEEE Access*, vol. 8, pp. 146284–146299, 2020.

[4] X. Sun, C. Wang, A. Sud, C. Xu, and L. Zhang, "MagicBrush: Image search by color sketch," in *Proc. ACM Multimedia Conf. (MM)*, Oct. 2013, pp. 475–476.

[5] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Color compatibility from large datasets," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–12, Jul. 2011.

[6] A. Manera, *Color Inspiration: 100 Color Palettes for Artists, Colorists, Crafters, Interior Designers, Hobbyists and More!* Independently Published, 2019.

[7] B. Yang, T. Wei, X. Fang, Z. Deng, F. W. B. Li, Y. Ling, and X. Wang, "A color-pair based approach for accurate color harmony estimation," *Comput. Graph. Forum*, vol. 38, no. 7, pp. 481–490, Oct. 2019.

[8] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–11, Jul. 2015.

[9] S. Liu and M. Pei, "Texture-aware emotional color transfer between images," *IEEE Access*, vol. 6, pp. 31375–31386, 2018.

[10] O. A. B. Penatti, E. Valle, and R. da Silva Torres, "Comparative study of global color and texture descriptors for web image retrieval," *J. Vis. Commun. Image Represent.*, vol. 23, no. 2, pp. 359–380, 2012.

[11] Shikisai Katsuyo Lab, *Color Scheme Handbook: Arranged by Style*. Tokyo, Japan: Pie Books, 2020.

[12] J. Ott, *1000 Ideas for Colour Schemes: The Ultimate Guide to Making Colours Work*. London, U.K.: Apple Press, 2016.

[13] N. Kuno, *Practical Color Combinations: A Resource Book With Over 2500 Sample Color Schemes*. Tokyo, Japan: Nippan IPS, 2018.

[14] Adobe. *Adobe Color*. Accessed: May 15, 2022. [Online]. Available: <https://color.adobe.com/create/image>

[15] *Color Palette Generator*. Accessed: May 15, 2022. [Online]. Available: <https://www.degraeve.com/color-palette/>

[16] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 703–715, Jun. 2001.

[17] Y. Chang, T. Iida, and N. Mukai, "Dominant color extraction method from natural images," *J. Inst. Image Electron. Eng. Jpn.*, vol. 44, no. 4, pp. 637–643, 2015.

[18] B. Berlin and P. Kay, *Basic Color Terms: Their Universality and Evolution*. Berkeley, CA, USA: Univ. California Press, 1969.

[19] N. Akimoto, H. Zhu, Y. Jin, and Y. Aoki, "Fast soft color segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8277–8286.

[20] J. Tan, J.-M. Lien, and Y. Gingold, "Decomposing images into layers via RGB-space geometry," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 7:1–7:14, Jul. 2017.

[21] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.

[22] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[23] S. Kim and S. Choi, "Dynamic closest color warping to sort and compare palettes," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–15, Aug. 2021.

[24] M. Bastan, H. Cam, U. Gudukbay, and O. Ulusoy, "BiVideo-7: An MPEG-7-compatible video indexing and retrieval system," *IEEE Multimedia*, vol. 17, no. 3, pp. 62–73, Jul./Sep. 2010.

[25] Demo. *Palette Based Image Recoloring Demo*. Accessed: Mar. 10, 2022. [Online]. Available: <https://recolor.cs.princeton.edu/demo/index.html>

[26] S. Ishihara, *Ishihara Test Chart Book, 38 Plate Book*. Tokyo, Japan: Handaya Shouten, 2011.



YOUNGHA CHANG (Member, IEEE) received the B.E. degree from Ewha Woman's University, in 1998, and the Ph.D. degree from the Tokyo Institute of Technology, in 2004. She became a Researcher, a Research Associate, and an Assistant Professor with the Tokyo Institute of Technology, in 2004, 2006, and 2007, respectively. Since 2012, she has been with Tokyo City University, Japan, where she is currently an Associate Professor with the Graduate School of Integrative Science and Engineering. Her research interests include image processing and color science. She is a member of ACM, SAS, ITE, and IPSJ.



NOBUHIKO MUKAI received the B.E., M.E., and Ph.D. degrees from Osaka University, in 1983, 1985, and 2001, respectively. He worked at Mitsubishi Electric Corporation and changed to worked as an Associate Professor with the Musashi Institute of Technology, in 2002. He has been a Professor with Tokyo City University, since 2007, and a Visiting Professor with the Institute of Industrial Science, The University of Tokyo, since 2020. He is currently a Professor with the Graduate School of Integrative Science and Engineering, Tokyo City University. His research interests include computer graphics and image processing. He is a member of ACM, SAS, VRSJ, IEICE, ITE, IPSJ, IIEEJ, JSUM, and JSAE.