Int. J. Human–Computer Studies 169 (2023) 102924

Contents lists available at ScienceDirect

International Journal of Human - Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Improving user experience of color palette extraction by using interactive visualization based on hierarchical color model

Raja Mubashar Karim [a,*,1], Taehong Jeong [b,1], Hyoji Ha [c], Jaejong Ho [c], Kyungwon Lee [c], Hyun Joon Shin [d]

[a] School Of Art Design And Architecture, National University of Sciences & Technology, Islamabad, Pakistan
[b] MAXST Co. Ltd., Gangnam-gu, Seoul, Republic of Korea
[c] Integrated Design Lab., Ajou University, Suwon-si, Gyeonggi-do, Republic of Korea
[d] Visual Computing Lab., Ajou University, Suwon-si, Gyeonggi-do, Republic of Korea

ARTICLE INFO

Keywords:
Authoring color palette
Hierarchical representation
Interactive visualization

ABSTRACT

Extracting color palettes from an image is a common practice used by artists in different visual domains. In this study, we introduce a novel tool for extracting color palettes from an image. Based on the hierarchical color model (HCM) (Jeong et al. (2019)), we develop a prototype user interface system comprising novel interactions and visualization. Our visualization is a node-link diagram tailored for the HCM; the accompanying interactions originate from the hierarchical structure. We evaluate our prototype system by performing a usability test and comparing it with contemporary alternatives for professional usage. The results from the user study validate the effectiveness of the presented approach. We also find a few user requirements that can be useful in the further development of the related tools. Moreover, we expect that the proposed interactive visualization can facilitate additional studies on the HCM. The prototype is available in the following URL: https://int-vis-hcm.web.app.

1. Introduction

Color palettes play a crucial role in visual domains ranging from illustration, graphic design, and fashion design to daily presentation design. They are also essential for many image manipulation techniques, such as image layer decomposition, recoloring, and colorization (Chang et al., 2015; Zhang et al., 2017; Cho et al., 2017; Xiao et al., 2020). Considering the existence of numerous online communities devoted to palette generation (Adobe, 2020; COLOURlovers, 2020; Coolors, 2020), it is not surprising that some people create them for their artistic expression and even for pleasure.

It is often preferable to generate palettes from an image because producing a palette from scratch requires talent and a significant understanding of color. Therefore, some of the existing tools allow users to generate a palette from a specific image, either automatically or manually. In automatic palette generation, it is crucial to supply users with an additional user interface for modifying the palette because the palettes desired by users are dependent on the users' intent and the context of the application. The aforementioned aspects encouraged us to develop an interactive approach that allows users to modify palettes based on the users' intent. In particular, we focus on an interactive palette creation methodology for novice users within the context of a specific image.

In this study, we propose a novel tool aiming at creating a positive user experience (UX) for extracting color palettes from an image. Our idea involves developing a user interface that allows users to investigate the colors in an image and generate a palette from the image. In particular, we adopt the hierarchical color model (HCM) (Jeong et al., 2019), which was developed recently. We believe that the "structural guidance" provided by the HCM can be beneficial to the task. In other words, a user can rely on the hierarchical structure when creating a palette. Potential solutions are embodied in the hierarchy, and the desired result can be obtained through systematic approaches. Through this *structured generation*, a beginner can extract a plausible palette. Additionally, a user can edit the palette by utilizing the constraints imposed by the hierarchy. Sometimes, even experts lose their context when manipulating a palette. We can prevent such situations using *guided manipulation* (Fig. 1).

The contributions of this study are summarized as follows: (1) We demonstrate that the HCM improves the UX of color palette creation. (2) We find a few user requirements, which will be valuable for

* Corresponding author.
E-mail addresses: mubashir.karim@sada.nust.edu.pk (R.M. Karim), thjeong@maxst.com (T. Jeong), hjha0508@ajou.ac.kr (H. Ha), hapsoa@ajou.ac.kr (J. Ho), kwlee@ajou.ac.kr (K. Lee), joony@ajou.ac.kr (H.J. Shin).

[1] Both authors equally contributed to this work.

https://doi.org/10.1016/j.ijhcs.2022.102924
Received 31 December 2020; Received in revised form 4 August 2022; Accepted 1 September 2022
Available online 9 September 2022

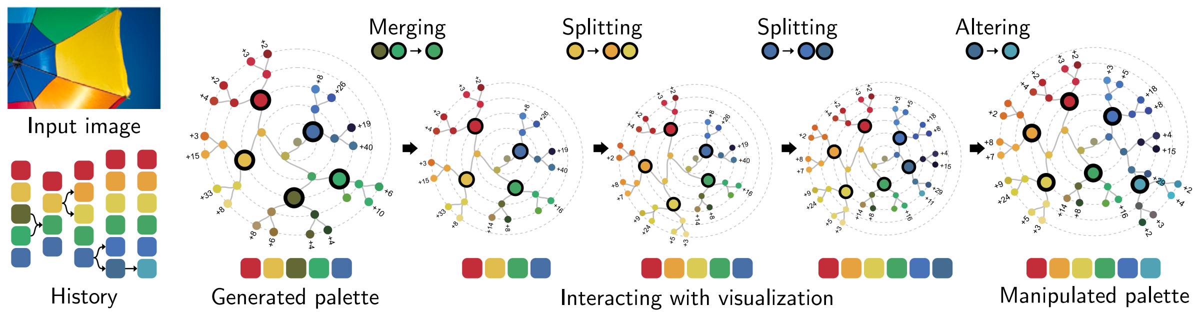

Fig. 1. A hierarchical color model (Jeong et al., 2019) of an input image is visualized in our approach. By interacting with the visualization, a user is able to extract a plausible palette from the image.

further improvement. (3) Our proposed interactive visualization helps to explore color hierarchy with significant potential.

This study is organized as follows: In Section 2, we review the related works, including those involving the HCM and perform a comparative study of the existing tools. Section 3 narrates the basic idea of our proposed user experience based on the HCM. In Section 4, we explain the prototype system that we developed as a proof of concept. In Section 5, we describe our user study, followed by the quantitative and qualitative feedback in Section 6. Section 7 concludes the study.

2. Previous work

In this section, we briefly discuss three groups of related works, followed by a review of the HCM. Next, we provide a comparative study on a few existing tools for authoring color palettes. By comparing their features, we estimate the range and limits of the current UX.

Interacting with color. Although user interfaces for color selection have changed little over the decades (Meier et al., 2004), a few studies have proposed novel user interfaces with an extended capability through expert interviews (Jalal et al., 2015; Shugrina et al., 2017). Concerning interacting with color palettes, two prior studies are noteworthy. One study enabled interactive palette exploration by imposing constraints on its graph-based representation (Mellado et al., 2017). The other introduced a unified framework for creating and editing different kinds of color palettes (Shugrina et al., 2019). Our study is in line with these efforts to improve interactions for color palette generation and manipulation.

Hierarchy visualization. We refer readers to a review on the visualization of hierarchical information (Schulz and Schumann, 2006). By rendering links, an explicit approach shows the relationship between nodes (Reingold and Tilford, 1981; Lamping and Rao, 1996). On the other hand, an implicit method focuses on visualizing quantities through space partitioning (Schulz et al., 2010). It is also common to classify the visualization techniques into either radial (Lamping and Rao, 1996) or axis-parallel arrangements (Reingold and Tilford, 1981). A notable study, 'Tree Colors,' (Tennekes and de Jonge, 2014) suggested a hierarchical visualization of the Hue-Chroma-Luminance color model. In the meantime, there have been efforts to develop interactive ways of exploring large scale hierarchical structures (Blanch et al., 2015; da Silva et al., 2019).

Color palette extraction. There are a few studies opted for data-driven approaches to solve palette-related problems (O'Donovan et al., 2011; Lin and Hanrahan, 2013; Kita and Miyata, 2016). The domain spans color compatibility (O'Donovan et al., 2011), aesthetic rating (Kita and Miyata, 2016), ordering (Phan et al., 2018), and so on. In these studies, the key idea was to exploit a palette data-set either created by humans or extracted from images. Studies on image manipulation proposed methods for obtaining the representative palettes of images. The majority are clustering-based methods, (Chang et al., 2015; Zhang et al., 2017) and the others are hierarchical approaches (Orchard and Bouman, 1991; Jeong et al., 2019).

Hierarchical color model. The HCM (Jeong et al., 2019) facilitates the extraction of palettes representing a specific image, which is the preliminary step in several image manipulation techniques. It is common sense that the more accurate the representation, the better the quality of the edited image. However, in such an unsupervised proxy generation problem, the single optimal answer is rarely found. Therefore, the HCM aimed to embrace all probable solutions in a hierarchical structure. Precisely, a hierarchy is built in a bottom-up manner, starting from the local estimates of the unknown global color model. As a result, a full binary tree is formed whereby the regional color distributions become leaves, and their mixtures become the internal nodes of the tree (see Fig. 1).

Existing tools

In this study, we compare the functionalities of conventional tools for creating and editing color palettes. Through this approach, we try to define the range and limits of the UX delivered by the existing tools. Because we focus on practicality, we only examine modern systems that are accessible to the public. Furthermore, we focus on the functionalities related to authoring color palettes. Note that in this study, we concentrate on the *discrete color palette*. For a complete overview, please refer to the study conducted by Shugrina et al. (2019).

Our comparison (see Table 1) includes a total of six applications: Adobe Color 4), Colordot (2020), Colormind (2020), COLOURlovers (2020), Coolors (2020) (the second one in Fig. 4), and Paletton (2020). Each column in Table 1 corresponds to each of the seven functionalities examined in this study. The first four functionalities are directly related to creating and editing palettes, whereas the rest give additional controls for convenience. Almost every tool provides a color picking interface; hence, we omitted it.

Table 1 shows a few noteworthy points. First, sometimes, the size of a palette is fixed. Even if the size is variable, its maximum is often limited to 5. However, up-to-date tools allow users to have much bigger palettes. We assume that this is because of the flexible user interface. Next, it is easy to find tools supporting image-based functionalities. Some of them automatically extract plausible palettes from an image, whereas others let one manually pick colors from the image. Some tools provide users with a tool for manipulating the results generated automatically.

Additionally, we found that *color harmony*, which imposes structural constraints between the colors in a palette, is widely supported. Some tools also provide *themes* for adjusting palette colors simultaneously. However, the capabilities for manipulating individual colors are limited to what is provided by the color pickers. Last, we note that some tools consider accessibility for people with color-blindness.

Although known tools provide practical means, there is room for improvement. For instance, it may be helpful having a variable-length palette without size limitation. Such a palette could expand or shrink to match a specific size. Moreover, it would be beneficial to have direct and coherent methods for editing a specific color of a palette extracted

Table 1
Comparison of functionalities of existing tools. Size – variable palette size (V) or fixed (F); Max – the maximum size of a palette; Image (automatic) – automatic palette extraction from an image; Image (manual) – manual selection of colors from an image; Harmony – global editing based on color harmony; Theme – global editing based on color theme; Accessibility – simulating color deficiency. (●: fully, ◐: partially, ○: not supported).

Tool	Size	Max	Image (automatic)	Image (manual)	Harmony	Theme	Accessibility
Adobe Color	F	5	●	●	●	●	●
Colordot	V	∞	○	○	○	○	○
Colormind	F	5	●	○	○	●	○
ColourLovers	V	5	◐	●	●	○	○
Coolors	V	10	○	●	●	●	●
Paletton	V	4	○	○	●	●	●

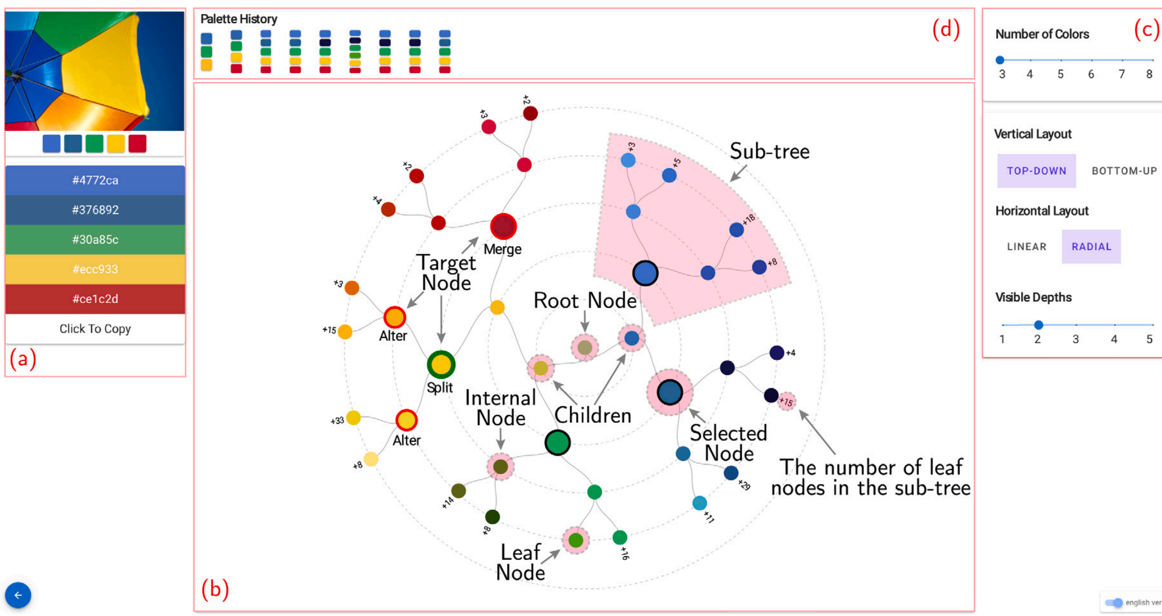

Fig. 2. A screenshot of our prototype(*HCM*) UI system: (a) The 'Input & Output' view, (b) the 'Chart' view, (c) the 'Settings' view, and (d) the 'History' view. Please note that the annotations are not included in the actual environment.

from an image. Such approaches may follow the same rule used in the extraction process. To provide these functionalities, we believe that there is a need for a new structural approach.

3. Use case scenarios

The adoption of the HCM offers significant merits that originated from "structural guidance". In this section, we explain its benefits within the context of a specific user story. Precisely, we regard a user who wants to author a palette representing a particular image and consider an imaginary tool that provides related functionalities.

Suppose that a user has selected an image from which the user wants to extract a palette. Given the image, the tool provides the user with an automatically generated palette. Assuming that the palette does not satisfy the user, they would like to do either of the following actions: (1) The user would like to change the parameters, if any, such as the number of colors in a palette so that the tool creates a new palette opportunistically. (2) The user would like to modify the palette to their taste using the interaction methods supported by the tool. In such processes of generation and manipulation, the incorporation of the HCM could introduce enhanced user experiences described as follows.

Structured generation. In the HCM, the hierarchy embraces the potential palettes, and a particular one can be selected automatically according to the size specified by the user. Therefore, the time taken to generate a palette is almost constant regardless of the palette's size. This efficiency guarantees that a user can change the size interactively. Additionally, the HCM automatically generates not only any palette for the given size but also the palette covering the colors in the image within the specified size. This *structured generation* allows even beginners to create desirable palettes fast and easily.

Guided manipulation. The probable solutions embedded in the hierarchy share structural relationships. By exploiting the parent–child constraints between nodes, users can reflect their intent directly. For example, the individual colors stored in a node as their mixture can be handled separately as their two children. Conversely, one can consider two sibling colors as a single color of their parent. In some cases, choosing one of the sibling colors and pruning the other, instead of combining them, might result in a more favored result. Such *guided manipulation* would be beneficial even to an expert by helping them not to lose their context when manipulating a palette.

One of the crucial points is that we also need a dedicated visualization to provide such a user experience. The color hierarchy of an image is a binary tree in which each node corresponds to a specific color. Although there are several ways to visualize the hierarchy, we display it explicitly because it contains a set of relationships between qualitative data. Note that the original study focusing on the HCM did not provide a visualization scheme. Although it is possible to develop a superior visualizing method by conducting an expert interview, in this study, we use a minimal design to actualize the proposed interactions.

In the following sections, we discuss the implementation of the aforementioned ideas of interaction and visualization in our prototype system as a tangible user experience. We then describe our user study (Section 5), which is designed to validate the proposed palette authoring tool.

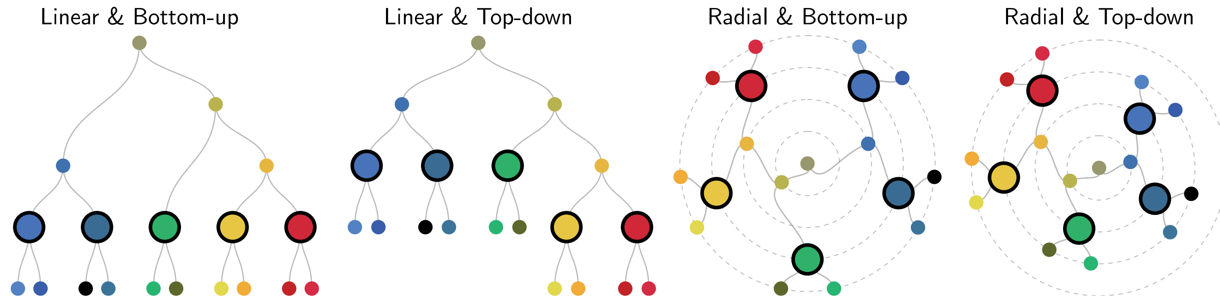

Fig. 3. A comparison of four layouts of our visualization. We provide two horizontal arrangements (the linear and the radial) and two vertical alignments (the bottom-up and the top-down) to accommodate different contexts and preferences.

4. Prototype system

In this section, we discuss the design and implementation of our prototype system. We start by describing the visualization of a given color hierarchy (Section 4.1). Additionally, we introduce the ways of interacting with the visualized structure (Section 4.2) to manipulate a generated palette. We conclude this section with a brief outline of the user interface design (Section 4.3) before proceeding with our discussion of the user study. Note that our system is merely a prototype whose purpose is to enable and ease the user study for validating the proposed concepts.

4.1. Color hierarchy visualization

As mentioned earlier, a color hierarchy comprises a set of nodes, each of which corresponds to a particular color. Some of them are in the palette; the others are not. We refer to such a node whose color is in the palette as a *selected* node. Additionally, a particular node with which a user currently interacts is called a *target* node hereafter.

We adopt a node-link diagram to visualize a given HCM explicitly. Each node in the diagram is rendered as a circle filled with the corresponding color. If any distinctions are needed, we change the radius of the circle and the color of the border. In Fig. 2, the selected nodes are magnified in size and have a black border, and the target nodes are distinguished in specific ways. We also utilize text labels to give more information. For instance, the interaction target has a label denoting the type of the corresponding operation. Additionally, if an internal node is drawn at the terminal, we specify the number of leaf nodes under the folded sub-tree.

As shown in Fig. 3, we array the nodes in two breadth-wise arrangements, namely linear and radial. The former naturally shows the spatial relationship between nodes (the left-hand side), whereas the latter is superior in terms of space efficiency (the right-hand side). In the radial arrangement, we depict concentric lines to ensure that the depth and height are perceived clearly. We also allow the user to have different depth-wise alignments of the nodes: bottom-up (the first and the third), and top-down (the second and the fourth). Each of the four layouts has pros and cons, and we let the user configure them according to the occasion (see Section 4.3).

When the user selects a layout setting, the actual positions of the nodes in 2D space are determined using the Reingold–Tilford algorithm (Reingold and Tilford, 1981), which helps us to minimize space usage. A color hierarchy often contains hundreds of nodes. Therefore, it is not easy to draw the entire hierarchy at once. To keep a diagram as compact as possible, we limit the depth visible after each selected node. Moreover, we allow the user to adjust the *visible depth* to explore the hierarchy.

In Fig. 9, we show the visualization instances corresponding to various styles of photographs taken from the MIT-Adobe FiveK dataset (Bychkovsky et al., 2011). Most of the previous studies presented an extracted palette along with the input image (similar to the one shown in Fig. 2.) Compared to them, our visualization expresses the colors of each image in a more articulated manner.

4.2. Palette manipulation interaction

We provide a user with four primitive operations, including *altering*, *splitting*, *merging*, and *reverting*. The original study (Jeong et al., 2019) proposed the *splitting* and *merging* operations, and we extended the capability by presenting the other two. We summarized the four operations and their properties in Table 2. We define the four operations as follows:

Altering. This operation changes a particular node to one of its children. Using this operation, a user may opt to choose the left or right child of the node. As a result, the user can fine-tune one of the palette colors to be more desirable.

Splitting. This operation breaks a target node into its two children. Precisely, the operation first removes a specific color from the palette and appends two children of the removed one. This operation is useful for users who want to have more detailed colors in the place of abstract ones.

Merging. This operation combines a specific node and its sibling into their parent. From the palette, the operation removes the color of the target node and that of its sibling, and it inserts the color of the parent instead. It allows users to handle similar colors at once.

Reverting. This operation is the inverse of *altering*. It undoes the designation of a particular node. The object of this interaction is the node whose sibling's color is not in the palette.

Combinations of the primitives cover all the possible interactions. Even so, it is efficient to combine frequent sequences of actions into a single one. Such compacted operations allow users to manipulate the palette more directly and intuitively. To define an *advanced* manipulation operation, we find the goal of a particular interaction considering the circumstance. Indeed, we can categorize all possible advanced operations into three groups based on the relationship between a target node and the currently selected nodes. Each group involves directly selecting an ascendant, a descendant, or the sibling of a selected node.

Ascendant-case. When an ascendant of any selected nodes is targeted, the selected nodes below the target are removed from the palette, and the target is added instead. For instance, if a user selects the root, only the color of the root remains in the palette. This simplifies a sequence of *merging* (or *reverting*) operations.

Descendant-case. When a user tries to interact with a descendant of any selected node, we recognize it as a series of *altering* operations so that the user can pick a specific node directly.

Sibling-case. A user might want to turn the result of an *altering* operation into that of a *splitting* operation on the parent node. This can be done through a series of one *reverting* operation and one *splitting* operation. Otherwise, the user can interact with the sibling that was opted out in the process of *altering*, and directly add it into the palette.

Table 2
The four primitive operations are summarized. In the second column which corresponds to the direction of change, ↓ (or ⇓) denotes the direction from a parent to its child (or children), and ↑ (or ⇑) denotes the opposite. The third column corresponds to the change of the size of the palette. The next two columns are the two operands required by each operation. The last column is the output of each operation.

Operation	Dir.	ΔSize	Operands	Result
Alter	↓	0	A selected node having children & One of its children	The child
Split	⇓	+1	A selected node having children	The children
Merge	⇑	−1	A selected node whose sibling is chosen & Its sibling	The parent of them
Revert	↑	0	A selected node whose sibling is not chosen & Its parent	The parent

4.3. User interface

We developed a user interface comprising distinct panels, including our interactive visualization (Fig. 2.) The implementation is a web application that has been deployed in the public domain: https://int-vis-hcm.web.app.

Input & output. It is inevitable to keep comparing the image and the palette to produce a better result. Therefore, we juxtapose the input image and the output palette at the most salient position. Additionally, we show the color hex codes of the colors in the palette and provide a way to copy them for future usage.

Chart. We render the visualized HCM of the input image in the middle of the user interface as large as possible. The rendering provides not only the interaction functionalities but also smooth transitioning animations. We draw the chart using d3.js (Bostock et al., 2011), which is fully featured for our requirements.

Settings. The user interface has a few configurable options. The foremost is the number of colors in an initial palette. One can generate a palette from an image by specifying the desired number of colors in the palette. The lower part of the interface controls the layout of the diagram in the chart (See Section 4.1 for further details.) The last option is the visible depth from the selected nodes. It helps when the nodes are too many to show up at the same time.

History

Each time the current palette changes, we record the latest result. By providing the records in a visual form, we render a continuous flow of the color palette extraction process. Moreover, by clicking each visual record, it is possible to roll back to a specific state. This feature helps to explore different possibilities.

The *History* has two states: one is the recording state, and another is the editing state. In the former state, the History saves any user-made changes of the *Chart*. The latter state applies as one of the records becomes the target of interaction. By picking a record except for the current one, a user lets the *Chart* be set to have the record as the current palette. Any records which had been saved after the picked one are considered as temporary and displayed dimmed. The user may compare the picked one and the temporary ones by hovering the mouse cursor which makes the record under the cursor undimmed. Also, it is possible to make one of the temporary records become the current palette by clicking. This editing state ends if the user clicks the picked record again or manipulates the *Chart* by other means. The exit from this state makes the temporary records be deleted and applies the recording state again.

5. User study

Our study aims to examine how *guided manipulation* based on the HCM improves the user experience in color palette extraction. For this purpose, we conducted a two-part user study. Here, we want to mention that our prototype is not a fully-featured palette extraction system; it is only meant for testing the usability of guided manipulation.

User study design. We designed a user study comprising two parts. In the first part of the study, we performed a usability test to evaluate the overall effectiveness of our prototype's user interface by obtaining *Qualitative Feedback* (Section 6.1) from the participants.

In the second part, we performed a comparative evaluation of our prototype with two commonly used free web-based applications currently available online and compiled a *Quantitative Feedback* (Section 6.2) out of it. For both parts of the study, we followed the think aloud protocol for all of the tasks enabling the participants to speak up their minds while they are performing those tasks, obtaining immediate feedback as well. We also carried out a pilot study before the formal user study to discover any issue with our user study design.

Participants. We recruited 35 participants (15 males and 20 females) who declared themselves not related to color-blindness. All participants are Industrial Design undergraduate senior year students. All of the 35 participants participated in both parts (usability testing and comparative evaluation) of the user study. All of them had taken computerized design-related courses. Therefore, all were knowledgeable about digital color and palettes. To ensure sincere responses, we paid the participants for their time and effort.

Experimental setup. We used two 27-inch displays (2560 × 1440 pixels) of the same model for our study. One screen showed the interface for performing a task and the other display showed the complementary information (if any) regarding the task. All the tools were displayed using the Google Chrome browser. Participants performed their tasks using an external keyboard and mouse.

5.1. Usability test

Our first study was on the usability of our visualization and interaction methods, as well as that of the overall user interface. In particular, we tested the user experience of the prototype and the effectiveness and intuitiveness of the proposed interactive visualization.

For this, we designed five tasks to be performed by each participant. Each task was followed by a detailed questionnaire to discuss their experience. After finishing all four tasks, we gave a post-testing questionnaire to each participant. The questionnaire was devised to obtain useful insights from the participants' overall experience with the tool. The five tasks are described as follows.

1. From a given image, the participant was asked to extract a five-colored palette using our prototype.
2. The participant was asked to perform *splitting* and *merging* operations in our prototype.
3. The participant was asked to perform *altering* and *reverting* operations in our prototype.
4. The participant was asked to perform an undo function (move/get back to previous palette state) via the *History* view.
5. The participant was asked to adjust each specific option available in the *Settings* view (as shown in the right-hand side of Fig. 2.)

The result from the detailed questionnaire of usability testing was compiled in the form of *Qualitative Feedback* in Section 6.1

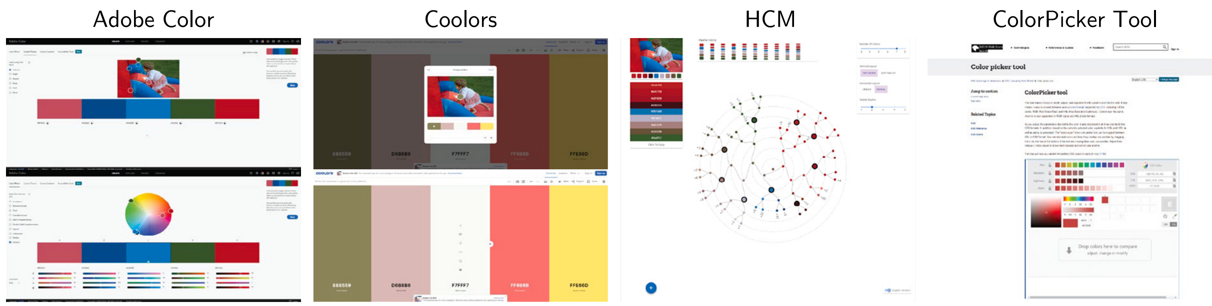

Fig. 4. Tools used in our comparative evaluation. In each task, Adobe Color (Adobe, 2020), Coolors (Coolors, 2020), and our prototype system are provided for the participants. Note that Adobe Color and Coolors present two different views for editing color palettes. For Task 3, ColorPicker Tool (Mozilla, 2020) is involved as an auxiliary application for a specific purpose.

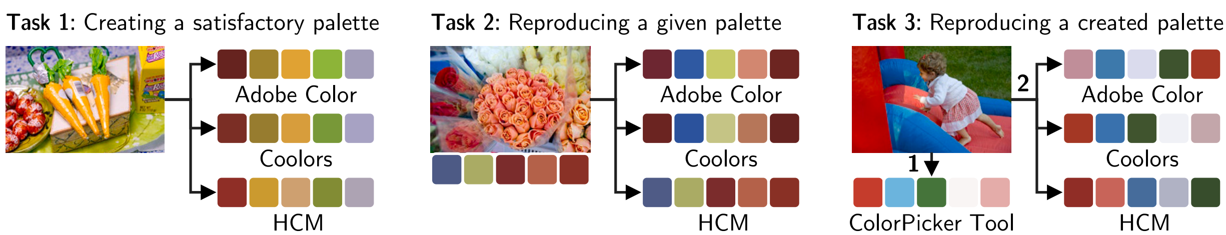

Fig. 5. A visual summary of the three tasks performed in our comparative evaluation. Note that Fig. 4 shows the main interface of three comparative tools along with the color picking tool used for this study.

5.2. Comparative evaluation

The second part of our user study focused on comparing two web-based palette authoring applications and our prototype. This comparative evaluation aims to evaluate our prototype's functionalities and user experience against frequently used alternates. To implement the experiment, we devised three tasks to understand user behavior while extracting a color palette from an image. It is to be noted that the HCM provides the most relevant colors, which sometimes might not be the most satisfactory ones for a user. For that reason, the participants are asked to extract a color palette in three different scenarios where the target palette is not necessarily the most relevant color palette.

As alternatives, we chose the following web-based applications: *Adobe Color* (Adobe, 2020) and *Coolors* (Coolors, 2020). These applications are used widely in real-world contexts because of their numerous features and high-quality interface design. The three tasks, which are performed by the participants using each tool, are described below. A visual summary is also provided in Fig. 5, showing how each task is executed.

1. *Creating a palette*: For this task participant was given an image from a pool of various images and was asked to extract a five-colored palette to their satisfaction.
2. *Reproducing a given palette*: Given an image and a corresponding five-colored palette, the participants were asked to reproduce the same palette using the given tool.
3. *Reproducing a user created palette*: The participants were first asked to make a five-colored palette representing the given image using a color picker (Mozilla, 2020), after which they were asked to reproduce the same palette using the given tool.

We conducted our test based on the Shackel usability model (Shackel, 2009). Although the original Shackel usability model consists of four aspects, we added the *satisfaction* aspect in our study because we believed that it was important for the user experience. Therefore, we evaluated five aspects of user experience: efficiency, satisfaction, attitude, flexibility, and learnability. In all usability aspects, a higher score means better performance except attitude. In the case of attitude, less score means better performance. Later in data analysis we flipped the attitude scores for ease of understanding.

After each task, the participants were given a questionnaire with a 5-point Likert scale (1 for strongly disagree, and 5 for strongly agree). Based on 5-point Likert scale data we compiled the *Quantitative Feedback* in Section 6.2.

Upon finishing all the tasks, the participants were asked to compare their experience with the tools. We also asked them to give their opinion regarding which tool they would prefer and why in *Post-test Interview* (Section 6.3).

5.3. Procedure

The user study consisted of five phases: (1) Introduction and brief about the study, (2) Training, (3) Demographic questionnaire, (4) Usability study, and (5) Comparative evaluation. Before the introduction, all participants were requested to provide informed consent of participation in our study involving color palette extraction from an image. Upon signing, we briefed all the participants regarding the purpose of the study.

As the first phase of the procedure, we provided them with an introduction on color palettes and how they can be extracted from an image using all three tools (*Adobe*, *Coolors* and *HCM)*. We used a few figures and diagrams to help the participants understand this process. Here we also explained the user interface and interactive feature of other two tools to the participants. After explaining the principles of our prototype to the participants, we navigated them towards our user interface and briefed them about the essential interaction methods of our prototype and how they can extract the color palette. We also showed participants a tutorial video (URL: https://rb.gy/yvkwck) of our prototype *HCM* for better understanding.

Before the formal user study, we let participants train themselves using the prototype and the other two selected tools. For this purpose, we provided them with a separate set of images which shared themes with the ones used in the main study. During the training process, the interviewers helped the participants understand the process. The participants were also allowed to ask questions if they felt confused at any point. This approach helped the participants understand the interactions and the components of the provided tools, which contains interactions of a unique style. Prioritizing the accuracy of the task, we asked the participants to respond as quickly and accurately as possible. Next, participants reported their necessary demographic information along with their experience using digital tools for color and palettes. Additionally, we requested the participants to self-report about color blindness before they started the formal study.

In the user study, the usability test always preceded the comparative evaluation. In the latter experiment, we assigned each participant a random order in which to use the three tools to nullify the effect of distribution of tools if any. Once the sequence was settled, it was maintained over the tasks. We provided each participant eight images, five for usability test and three for comparative evaluation randomly chosen from the fixed pool of photographs. Each image in the subset was paired with each task and fed as input to all the tools. Note that all the images are taken from the MIT-Adobe FiveK dataset (Bychkovsky et al., 2011), which contains various kind of scenes including flowers, foods, landscapes, and so on. Among the 5,000 images, we carefully picked out those containing fairly vivid and simple scenes.

6. Results & discussion

In this section, we review the results of the study and discuss the feedback through a quantitative analysis. We first summarize the qualitative user feedback regarding our proposed prototype system gathered from *Usability Test*. Next, we perform a quantitative comparison of our prototype with the web-based palette authoring tools based on the results from *Comparative Evaluation*. Finally, we discuss ways to improve the prototype based on the users' post-test feedback.

6.1. Qualitative feedback

Overall experience. In the usability test feedback, 29 participants said that the overall usage was easy to understand the process of color palette generating process from an image was unique and efficient. Participants emphasized how it enabled them to explore different color palettes within a given image with ease: "The process was easy to understand and once it was explained in the training session, I was able to generate a color palette without any further assistance." Another one said, "I could explore the colors within the given image smoothly and also be creative at some point."

Splitting & merging. Participants said that both of the interactive operations (split and merge) were intuitive and useful in exploring the colors in a specific visualization. Six participants said that it took them two to three attempts on understanding the *merge* function, but once they understood, it was very easy to move through different levels of color hierarchy while generating the color palette. One participant said that the *merging* operation was easy to understand. However, its visualization is a bit fast. For example, " I understood the function *merge* as the word merge itself is very intuitive and you can guess what it will be doing. But when it is done, it is difficult to visualize and understand what exactly happened on the screen. Sometimes you get lost in different levels of color hierarchy resulting in confusion between the colors. Eventually, it actually grows on you after you perform it two to three times."

Altering & reverting. All the participants said that the altering and reverting operations were beneficial, considering the nature of the visualization: "One needed to explore the colors within the hierarchical visualization by going back and forth to get the desired color. As a result, both operations were intuitive and convenient." Five participants also said that they did not understand what the *altering* operation meant. One of them said, "I did not understand what altering meant and how it worked. I was confusing *alter* and *split* most of the time. Many of the times I intended to split the color but I ended up altering a color."Another participant said "In brief, listening about altering function made sense and seemed to be integral to the palette authoring process but while performing it was difficult to demonstrate it."

History. All the participants said that it was useful to have a *History* of the palette extraction process. It kept a record of the users' previous palettes for comparison purposes. One of the participants said, "It was good to have a record of one's process. You can see all your palettes in a linear fashion. It helped me to compare my previous palettes with the one in process to accomplish a better final outcome." Another participant acknowledged the effectiveness of *History* view by saying "It is actually very effective. Although I can go back to my previous palettes using other interactive functions but somehow this seemed easy to go back or undo your work."

Size of the initial palette. When we asked the participants about the *Settings* operations, they said that these menu operations enhanced the user experience with the visualization as per their requirements. For example, the slider for the *number of colors* could be used if participants wanted more of an automatic selection by the tool. It may save processing time. Although they agreed that the slider was useful in selecting the *number of colors* in a palette, 17 of the participants said that they preferred to explore using the visualization instead of getting an automatic extraction result from the given options.

One participant acknowledged the usefulness of choosing a different number of colors in a palette. He commented, "I liked the flexibility of choosing a certain number of colors in a palette in the proposed tool. Most of the tools had fixed the number of colors in a palette (five in most of the cases.) I often needed three or more than five-colored palette instead of a five-colored palette." A participant said, "Giving a choice in the number of colors in a palette to the user is something new that many available tools do not provide in their palette authoring process."

Layout. 27 participants preferred the default *(Radial, Top-Down)* layout of the visualization. Eight participants said that the linear layout helped them to understand the hierarchy compared to the radial layout. But when you are in deep sub levels of color hierarchy, sometimes the colors overlap each other and confuses the user. However, when they interacted with the hierarchy to manipulate the color, they said they felt more at ease using the radial layouts.

Visible depths. 31 participants said that they like the default setting of two-levels *visible depths* as it allows them to be more focused while performing their task. A participant said that he is satisfied with the default settings, it keeps him more focused. At times when he needs to look into sub levels to find a color, for that he does play around with the *visible depths* options but most of the time uses the default settings. Other participants said that they preferred two *visible depths* as this helped them interact with the visualization and explore more colors. According to them, having more *visible depths* made the process difficult, and they could not remain focused because of the many colors that were displayed. Four participants said they preferred using more depths as this allowed them to see a variety of colors to explore. For example, a participant said "with four *visible depths*, I had more freedom to explore and observe the available colors in the next depths."

6.2. Quantitative feedback

To analyze the quantitative feedback from the comparative evaluation (Section 5.2), we perform the Pearson's Chi-Squared test. We performed this test to find the statistically significant relationship among the given variables. Note that this test applies to our data from a 5-point Likert scale—basically an ordinal scale.

For each question in each task, our hypothesis is that *HCM* will outperform the other tool - *Adobe Color* (or *Coolors*) or at least will be at par with them. Because the proposed system is a prototype comprising a minimal set of necessary features, we expect the other two tools may perform better in terms of certain usability aspects. Therefore, if the null hypothesis is not rejected, we see that our system's performance is acceptable regarding a specific usability aspect under a certain task. On

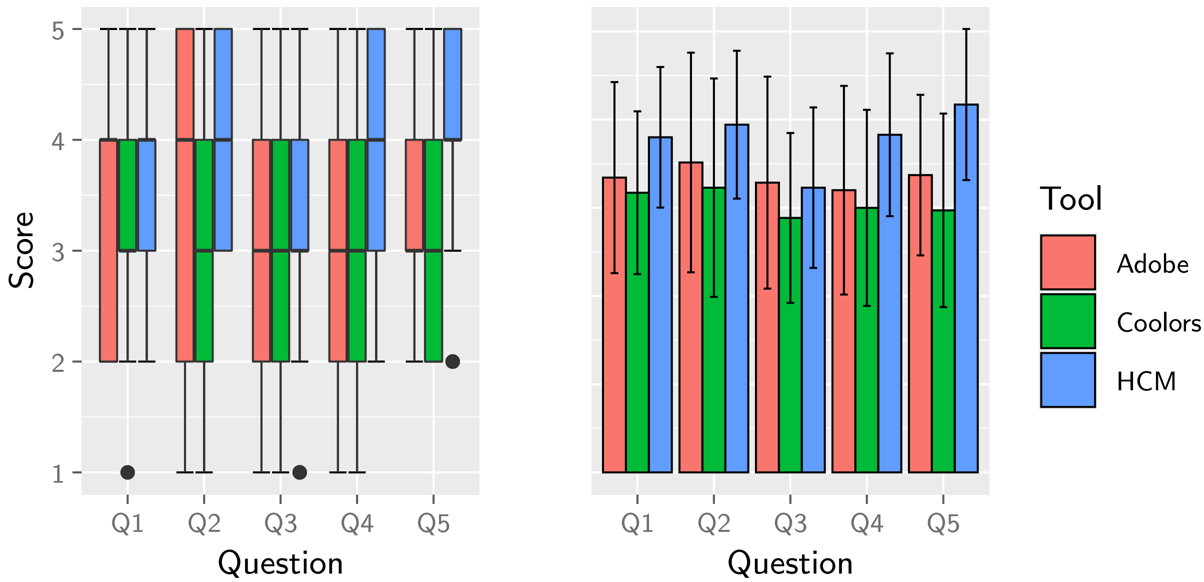

Fig. 6. Plots showing the descriptive statistics of the scores reported in Task 1. The boxplot (left) depicts the quartiles and outliers; the barplot (right) presents the means and standard deviations. The x-axis corresponds to questions: Q1—Efficiency; Q2—Satisfaction; Q3—Attitude; Q4—Flexibility; Q5—Learnability. The y-axis is a 5-point Likert scale.

Table 3
Descriptive statistics of Task 1. Each row corresponds to a specific tool, and columns give the mean μ and the standard deviation σ of the scores reported by the participants for each question.

Tool	Q1 – Efficiency		Q2 – Satisfaction		Q3 – Attitude		Q4 – Flexibility		Q5 – Learnability	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Adobe Color	3.26	1.146	3.31	1.323	3.03	1.317	3.57	1.037	3.49	1.040
Coolors	3.46	1.067	3.40	1.193	2.89	0.993	3.09	1.173	3.11	0.932
Ours	3.94	0.838	4.06	0.906	3.49	0.887	4.26	0.741	3.83	1.014

Table 4
Test statistics of Task 1. From the top, Pearson χ^2 and asymptotic significance level p are presented. The p-value is marked in boldface if the null hypothesis is rejected ($p < 0.05$).

	Adobe Color vs. Ours					Coolors vs. Ours				
	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
χ^2	9.7293	9.4826	7.4600	11.5940	3.0294	16.0000	12.5455	7.0000	19.4797	9.6987
p	**0.045**	0.050	0.113	**0.009**	0.387	**0.043**	**0.014**	0.136	**0.001**	**0.046**

the contrary, if the alternative hypothesis H_a, "There is a statistically significant difference", is accepted, we further compare descriptive statistics such as mean and median. After that, we could say that one tends to perform better than the other.

Task 1. A chi-squared test showed that *HCM* had a significant effect on *Adobe* in two out of five usability aspects. Chi-squared and the p value for usability aspects are; Q1—Efficiency ($\chi^2 = 9.72$, $p = 0.045$) and Q4—Flexibility ($\chi^2 = 11.59$, $p = 0.009$).

Although *HCM* performed better in other three usability aspects (Q2—Satisfaction, Q3—Attitude and Q5—Learnability) there were no statistics that described the significant difference among them. We found that *HCM* outperformed *Coolors* on four out of five usability aspects. Chi-squared test and the p value with respect to those aspects are; Q1—Efficiency ($\chi^2 = 9.84$, $p = 0.043$), Q2—Satisfaction ($\chi^2 = 12.54$, $p = 0.01$), Q4—Flexibility ($\chi^2 = 19.47$, $p = 0.001$) and Q5—Learnability ($\chi^2 = 9.69$, $p = 0.04$) (see Tables 3–5). . No significant difference was found in terms of Q3—Attitude (see Fig. 6).

Task 2. The results of Task 2 are significant but slightly different in terms of usability aspects compared to those of Task 1, given the test statistics presented in Table 6. Regarding the comparison between *HCM* and *Adobe Color*, we found significant difference in Q2—Satisfaction ($\chi^2 = 9.53$, $p = 0.04$), Q3—Attitude ($\chi^2 = 10.49$, $p = 0.03$), Q4—Flexibility ($\chi^2 = 11.56$, $p = 0.02$) and Q5—Learnability ($\chi^2 = 8.92$, $p = 0.06$). No significant difference was found among them for Q1—Efficiency (see Fig. 7).

In comparison of *HCM* and *Coolors*, *HCM* performed better in Q1—Efficiency ($\chi^2 = 9.51$, $p = 0.02$), Q2—Satisfaction ($\chi^2 = 11.21$, $p = 0.01$) and Q4—Flexibility ($\chi^2 = 10.69$, $p = 0.03$).

Task 3. In Task 3 we found significant difference in Q2—Satisfaction ($\chi^2 = 11.69$, $p = 0.02$), Q3—Attitude ($\chi^2 = 11.05$, $p = 0.02$) and Q4—Flexibility ($\chi^2 = 9.61$, $p = 0.04$). Both tools performed well in Q1—Efficiency and Q5—Learnability but with no significant difference (see Tables 7–9).

While comparing *HCM* with *Coolors*, likert scores revealed a significant difference in Q1—Efficiency ($\chi^2 = 10.44$, $p = 0.03$), Q2—Satisfaction ($\chi^2 = 12.86$, $p = 0.01$) and Q4—Flexibility ($\chi^2 = 10.60$, $p = 0.01$).

Discussions. By viewing the result from Task 1, 2 and 3 collectively, we note that the participants tend to see *HCM* is better than the others in terms of majority of the usability aspects. The first task asked the participants to create an image-representing palette satisfying themselves, and there is no exact answer in this case. At the same time, *HCM* provides the potential palettes well-designed already to represent a specific image. Therefore, using *HCM*, the participants can produce satisfactory results swiftly. Although there is an exact answer in Task 2, a similar situation arises. Because the given target palette faithfully represents the corresponding image, the participants have to search only

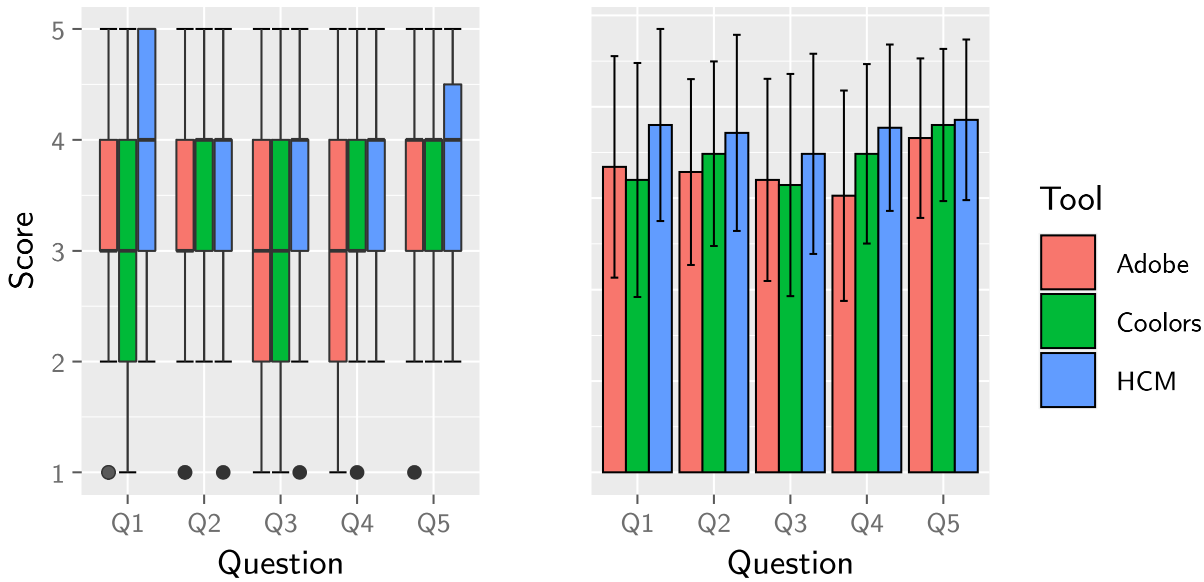

Fig. 7. Plots showing the descriptive statistics of the scores reported in Task 2. The boxplot (left) depicts the quartiles and outliers; the barplot (right) presents the means and standard deviations. The x-axis corresponds to questions: Q1—Efficiency; Q2—Satisfaction; Q3—Attitude; Q4—Flexibility; Q5—Learnability. The y-axis is a 5-point Likert scale.

Table 5
Descriptive statistics of Task 2. Each row corresponds to a specific tool, and columns give the mean μ and the standard deviation σ of the scores reported by the participants for each question.

Tool	Q1 – Efficiency		Q2 – Satisfaction		Q3 – Attitude		Q4 – Flexibility		Q5 – Learnability	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Adobe Color	3.54	0.980	3.29	1.017	3.03	1.124	3.03	1.150	3.51	0.887
Coolors	3.31	1.105	3.34	0.998	3.14	1.216	3.20	1.106	3.80	0.833
Ours	3.91	0.951	3.86	0.974	3.83	1.014	3.86	0.845	4.09	0.742

Table 6
Test statistics of Task 2. From the top, Pearson χ^2 and asymptotic significance level p are presented. The p-value is marked in boldface if the null hypothesis is rejected ($p < 0.05$).

	Adobe Color vs. Ours					Coolors vs. Ours				
	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
χ^2	4.7323	9.5333	10.4945	11.5692	8.9255	9.5165	11.2175	3.2756	10.6390	0.9961
p	0.192	**0.049**	**0.033**	**0.021**	0.063	**0.023**	**0.011**	0.513	**0.031**	0.802

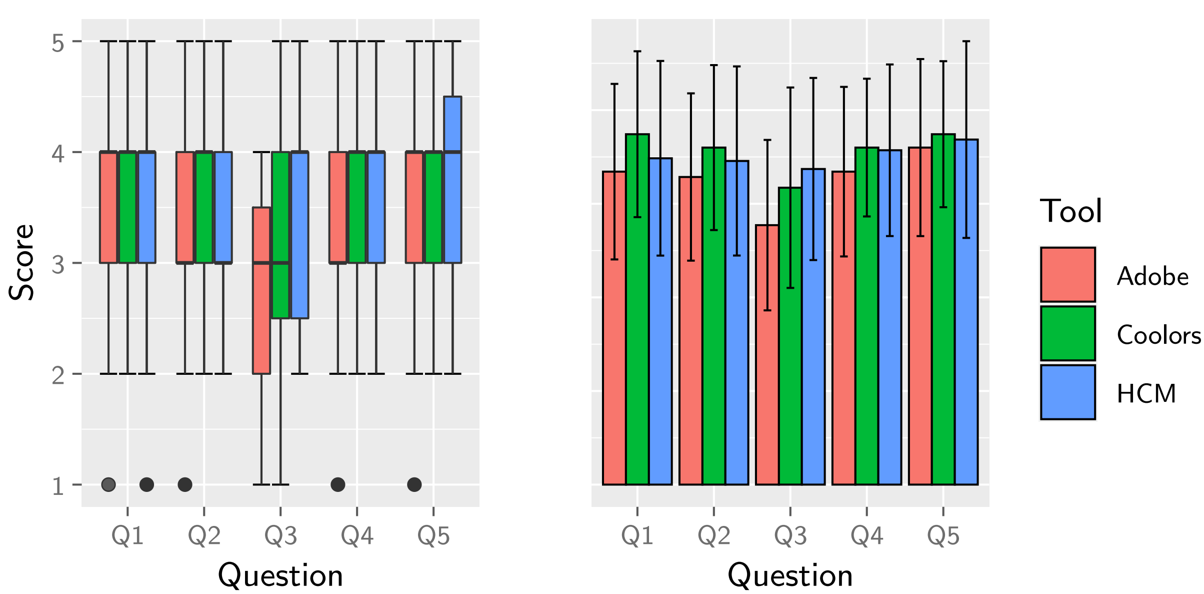

Fig. 8. Plots showing the descriptive statistics of the scores reported in Task 3. The boxplot (left) depicts the quartiles and outliers; the barplot (right) presents the means and standard deviations. The x-axis corresponds to questions: Q1—Efficiency; Q2—Satisfaction; Q3—Attitude; Q4—Flexibility; Q5—Learnability. The y-axis is a 5-point Likert scale.

Table 7
Descriptive statistics of Task 3. Each row corresponds to a specific tool, and columns give the mean μ and the standard deviation σ of the scores reported by the participants for each question.

Tool	Q1 – Efficiency		Q2 – Satisfaction		Q3 – Attitude		Q4 – Flexibility		Q5 – Learnability	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Adobe Color	3.34	0.938	2.94	0.998	2.77	0.910	3.23	0.877	3.60	0.946
Coolors	3.06	0.802	2.80	0.964	2.80	1.079	3.43	0.739	3.74	0.780
Ours	3.54	1.039	3.69	1.022	3.51	0.887	3.89	0.867	3.80	0.964

Table 8
Test statistics of Task 3. From the top, Pearson χ^2 and asymptotic significance level p are presented. The p-value is marked in boldface if the null hypothesis is rejected ($p < 0.05$).

	Adobe Color vs. Ours					Coolors vs. Ours				
	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
χ^2	2.3160	11.6984	11.0526	9.6149	3.4942	10.4410	12.8610	7.8439	10.6027	3.6250
p	0.678	**0.020**	**0.026**	**0.047**	0.479	**0.034**	**0.012**	0.097	**0.014**	0.305

Table 9
Raw data of our study. All 35 participants (15 male and 20 female) answered every question of the three tasks.

		Task 1					Task 2					Task 3				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Q1	Adobe Color	2	9	6	14	4	0	8	7	15	5	1	6	10	16	2
	Coolors	2	3	13	11	6	0	11	8	10	6	0	10	13	12	0
	Ours	1	1	4	22	7	0	2	6	15	12	1	6	6	17	5
Q2	Adobe Color	4	6	8	9	8	2	5	12	13	3	1	13	10	9	2
	Coolors	1	10	5	12	7	0	7	15	7	6	2	13	11	8	1
	Ours	0	1	10	10	14	0	5	4	12	14	0	6	6	17	6
Q3	Adobe Color	6	6	9	9	5	2	11	10	8	4	2	13	11	9	0
	Coolors	3	9	13	9	1	3	9	8	10	5	3	13	9	8	2
	Ours	1	3	12	16	3	1	7	5	15	7	0	6	12	14	3
Q4	Adobe Color	0	6	11	10	8	4	8	8	13	2	1	5	16	11	2
	Coolors	4	7	9	12	3	1	9	13	6	6	0	1	22	8	4
	Ours	0	1	3	17	14	0	2	9	9	15	0	4	9	17	5
Q5	Adobe Color	0	8	8	13	6	1	2	14	14	4	1	2	13	13	6
	Coolors	1	8	14	10	2	0	2	10	16	7	0	1	13	15	6
	Ours	0	4	9	11	11	0	1	10	14	10	0	4	8	14	9

the representative colors. The best possible colors well-representing the image are provided in *HCM*, and therefore, the participants can pick them out without much effort. In all three tasks, the participants might utilize either the interactive operations or the widget controlling the number of colors. Being provided the two different ways of completing goals, they consider *HCM* is much flexible.

Compared to the former two tasks, Task 3 yields interesting but somewhat expected results (See Fig. 8). In particular, we did not find any significance against *Adobe* and *Coolors* in terms of Q1—Efficiency and Q3—Attitude, respectively. These two results originate in the design of this task in which the participants define their own single solution (using a separate tool) and replicate it via each tool. Unlike *HCM*, the other two tools allow the participants to choose any of all the possible colors. Thus, they can replicate their goal regardless of whether the solution well represents the given image, or not. On the contrary, using *HCM*, the participants are only able to select highly relevant colors. This difference causes the resulting palette from *HCM* being considered less satisfactory, and therefore the participants reported not significant scores in Q1–Efficiency (for *Adobe*) and Q3–Attitude (for *Coolors*). Specifically, using *HCM*, it took too much time to replicate the solution, and sometimes it was not possible. However, these drawbacks would be resolved if we incorporate a color picker, which is omitted deliberately, into our prototype system.

6.3. Post-test interview

After finishing all the tasks, we asked the participants to give us their opinion regarding each tool (in a comparative manner), which tool they preferred, and the reason behind their preference.

Based on the circumstances, the participants had different preferences among the tools. The participants who preferred to use *HCM* stressed our tool's supports for choosing, exploring, and experimenting with various colors within an image. Our proposed visualization and its novel interactive methods allowed the participants to explore and extract various colors from an image. One participant said, "*HCM* was good and comprehensive for exploring colors especially because of the hierarchical representation of colors. It allowed me to explore colors in depth because of its visualization. On the other hand, *Adobe Color* (Adobe, 2020) and *Coolors* (Coolors, 2020) failed in this area. They tended to limit my interaction in terms of exploring colors as you are eyeballing the image and selecting a color based on your instincts." Other participants said, "If the goal was to explore the hierarchical structure of color extracted from an image, *HCM* would be the best tool. However, if one wanted to extract colors from an image and make adjustments, I would prefer the other two tools, or I would say may be a combination of all three of them.", "I preferred *HCM* in terms of exploring the hierarchical structure of colors extracted from an image"., and "The interactive operations were precise for this kind of process. Each interactive operation helped me to focus on a specific color, which I wanted to explore further. This idea of color hierarchy extracted from an image is novel and exciting". Another notable comment was, "Creating a color palette from a hierarchy of color extracted from an image was unique, and the exploration process was significantly systematic".

One of the highlights of *HCM* mentioned by the participants was the flexible nature of the tool both in interactive functions and in overall interface of the tool. One said "*HCM* allows us to move back and forth in process. We can easily move to our previous steps if we want

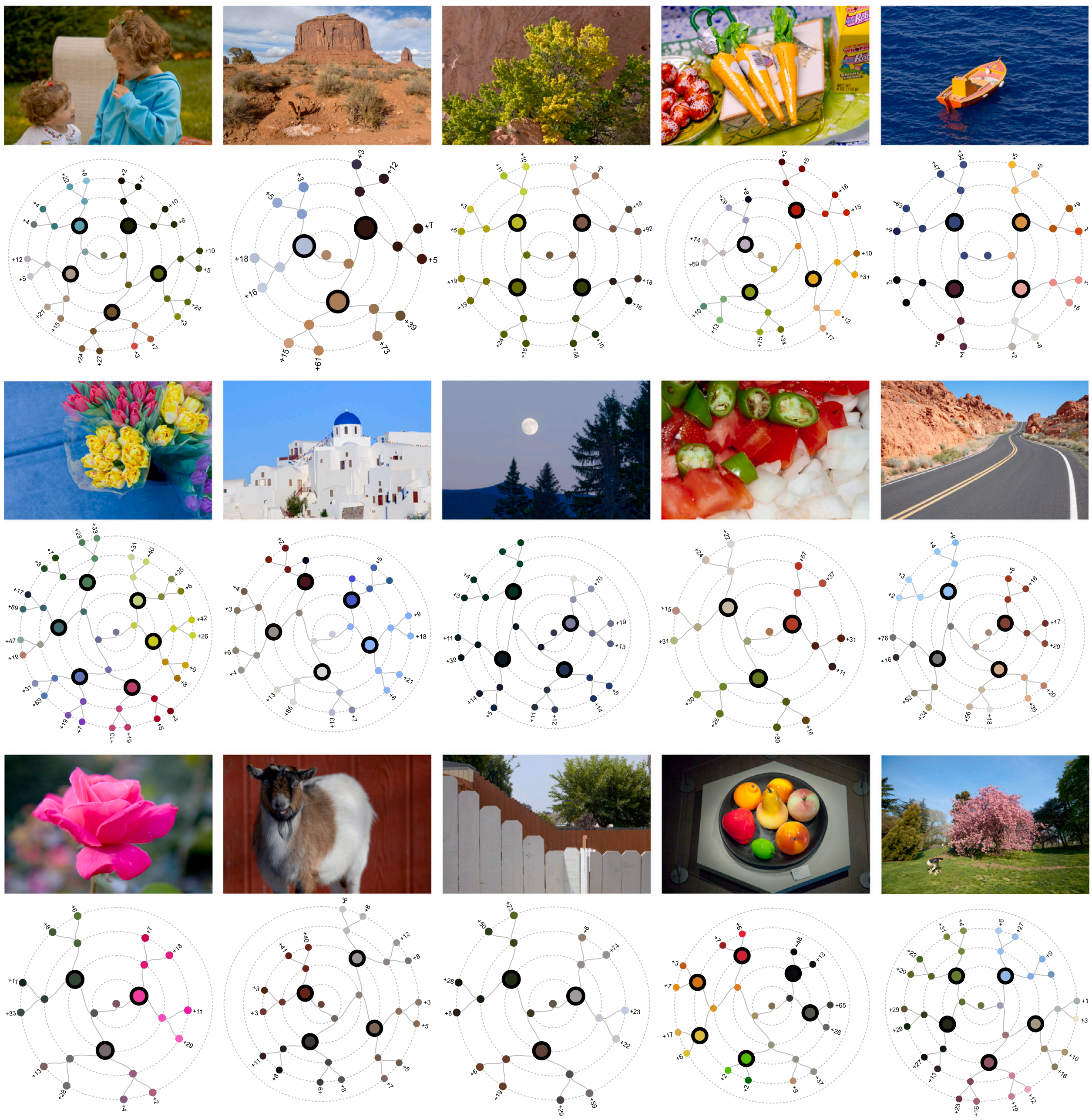

Fig. 9. Images with different contents and their corresponding color hierarchies are shown here. Each model is visualized in our method, and the layout is fixed to be radial & top-down. Note that not all the nodes are presented here to make diagrams more compact. The numbers attached at the terminal nodes indicate the number of leaves below the folded sub-tree. The highlighted nodes correspond to the color palette.

to. On the contrary both other tools does not allow this type of user experience”.

Some participant also mentioned about the flow of color generating process. A participant said, “ I like the overall process of *HCM* as compared to other two tools. I found the process of both tools (*Adobe* and *Coolors* was a little destructive in nature, especially when you navigate between their tabs”. Another participant said “ I had no clue when I navigated from one tab to the next because the reference image was not there any more to work with”. They also said the idea of having the *History* of the entire process was significantly helpful.

The participants who preferred the other tools said that they wanted more freedom in terms of color selection and adjustments. For example, one said, “Choosing the desired color was easier for me in *Adobe Color* and *Coolors*. *HCM* did not allow me to choose the desired color freely. It took several steps to do that. If you adjusted the selected color, you could not do that in *HCM*.” Another participant commented, “There was no way to adjust the colors after selecting them in *HCM*.” There was also feedback: “I wish I could delete any color of my choice from the palette directly. I found it very hassling to perform two to three steps to replace or delete a particular color.”

Generally, the participants mentioned a few usability nuances for each comparative tool. Participants also commented on specific features, such as *visible depths* in their feedback. Some said that when you are having more(+four) number of *visible depths* you often find it difficult reading into different colors as the lower level color nodes often overlap each other.

Our goal was to propose an interactive visualization with novel operations that could improve the user experience for color palette extraction from an image. Most of the participants agreed upon the effectiveness of our novel interactive user experience. Therefore, we concluded that the majority of the feedback regarding our proposed prototype was significantly positive. Participants also gave some suggestions, which could help us improve the user experience. Their ideas are summarized as follows:

Extended interaction. There was a suggestion regarding the selection of the *parent* and *sibling* nodes simultaneously when manipulating. In the current form, participants could only select either of them at the same time. Other participants liked the idea of picking a color from the input image as in *Adobe Color*. They commented that this interaction would improve the user experience.

Interaction on a part of an image. Some participants wanted to see which parts of the image corresponded to the selected color for better understanding. Additionally, there was a need for the visualization of colors only from a specific part of the image. Some participants even wanted to build a color hierarchy from only the designated part of the image.

Visualization control. A few participants said that the node size is not ideal for comparing their colors, especially those of the nodes in the sub-trees. This need can be satisfied by having adaptive control over the *visual depths*. In other words, users can focus only on the sub-tree of the selected node by minimizing the *visual depths* of the remaining nodes.

Although all the suggestions of the participants are reasonable, we have to mention that it is difficult to incorporate some of the proposed features:

1. directly picking a color from the input image as a way of interacting with the visual
2. visualizing correspondence between pixels and color palette nodes

The key challenge is the fact that the *HCM* does not retain the relationship between pixels and colors. Meaning, the spatial information is lost right after the local color distributions are estimated. This set of color space densities may describe each pixel's likelihood to fall into one of them. However, it is unnatural to determine which pixel exactly belongs to which of the distributions. We thus conclude that we need to extend the *HCM* to conserve the spatial context. Such a new model, which is one of our on-going work, would be sufficient to provide the above features.

7. Conclusion

In this study, we discuss a novel palette authoring tool with a better user experience, which originates from the *HCM*, for extracting color palettes from a given image. After reviewing prior studies, we suggest that structural guidance would be beneficial for the task. To confirm our proposition, we implemented the prototype system to conduct the user study. From the feedback, we established that our interactive visualization is easy to learn and convenient to use. The users also stated that based on their user experience the proposed tool is superior to existing ones. Based on these results, we expect that integrating the proposed functionalities with existing workflows will improve their UXs. It is noteworthy that our prototype was superior to the fully-featured tools at most of the occasions.

We conclude this study by presenting a few limitations and future directions. First, we adopted the *HCM* without any modification. To extend our tool for authoring an arbitrary color palette, we need to re-purpose *HCM,* which was proposed originally for image manipulation. Next, the proposed visualization of color hierarchies did not originate from expert feedback. We would like to conduct interviews with professionals to improve the visualization further. Finally, our system might not be enough for color palette authoring because it is only a proof of concept. The users requested additional features, such as direct color picking, theme modification, color harmonization, and reordering palettes. By incorporating such functionalities, we expect that the prototype will become a ready-to-use tool.

CRediT authorship contribution statement

Raja Mubashar Karim: Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Visualization. **Taehong Jeong:** Conceptualization, Methodology, Software, Writing – original draft, Visualization. **Hyoji Ha:** Conceptualization, Formal analysis, Investigation, Visualization. **Jaejong Ho:** Conceptualization, Software, Visualization. **Kyungwon Lee:** Conceptualization, Project administration, Supervision. **Hyun Joon Shin:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Questionnaire

The questionnaires used in the usability study, the post usability study, and the comparative evaluation are as follows. For each task in the comparative evaluation, participants were asked to complete a questionnaire using a 5-point Likert scale (1 for strongly disagree, and 5 for strongly agree).

Usability test - task 1.

1. Why did you prefer using the *Settings* options instead of the *Chart* (node-link diagram)? OR vice versa.
2. How easy or difficult was it to use the tool?

Usability test - task 2.

1. Do you understand the *split* and *merge* functions and how they work?
2. Probe: How was the experience of manipulating a color palette using *split* and *merge* functions and was there anything unclear about their functionality?
3. How easy or difficult was it to perform *split* and *merge* functions?

Usability test - task 3.

1. Do you understand the *alter* and *revert* functions and how they work?
2. Probe: How was the experience of manipulating a color palette using *alter* and *revert* functions and was there anything unclear about their functionality? How easy or difficult was it to perform *alter* and *revert* functions?

Usability test - task 4.

1. Does the *History* view help in your color palette generating process?
2. Do you prefer to use *History* to return to the previous state in the process of visualization interaction? Or you prefer to go back to the previous state using a main view interaction?

Usability test - task 5.

1. What difference do you feel when using the *Settings - number of colors* operation compared to manually generating palettes from the visualization?
2. Which *Layout* do you prefer and why?
3. *Visible depths*: how many do you prefer (less or more number of depths) and why?

Usability test - post.

1. How would you describe your overall experience with the tool?
2. What did you like the most about using this tool? What did you like the least, or what caused your frustration?
3. What, if anything, surprised you about the experience?
4. If you could change one thing about the tool, what would it be? Why?

Comparative evaluation - task 1.

1. Was the tool efficient enough to generate a five-colored palette of your liking?
2. Were you satisfied with the end result?
3. Was there any discomfort/frustration when performing the task?
4. Does the tool allow you to choose your method of interaction?
5. Is the tool easy to learn and understand?

Comparative evaluation - task 2.

1. Was the tool efficient enough to match the palette as shown in the given image?
2. Were you satisfied with the end result?
3. Was there any discomfort/frustration when performing the task?
4. Does the tool allow you to choose your method of interaction?
5. Is the tool easy to learn and understand?

Comparative evaluation - task 3.

1. Was the tool efficient enough to re-generate a five-colored palette that you generated using the color picker?
2. Were you satisfied with the end result?
3. Was there any discomfort/frustration when performing the task?
4. Does the tool allow you to choose your method of interaction?
5. Is the tool easy to learn and understand?

References

Adobe, 2020. Adobe color: Color wheel, a color palette generator. URL: https://color.adobe.com. (Online; Accessed 23 June 2020).

Blanch, R., Dautriche, R., Bisson, G., 2015. Dendrogramix: A hybrid tree-matrix visualization technique to support interactive exploration of dendrograms. In: 2015 IEEE Pacific Visualization Symposium. PacificVis, IEEE, pp. 31–38.

Bostock, M., Ogievetsky, V., Heer, J., 2011. D^3 data-driven documents. IEEE Trans. Vis. Comput. Graphics 17 (12), 2301–2309.

Bychkovsky, V., Paris, S., Chan, E., Durand, F., 2011. Learning photographic global tonal adjustment with a database of input output image pairs. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition.

Chang, H., Fried, O., Liu, Y., DiVerdi, S., Finkelstein, A., 2015. Palette-based photo recoloring. ACM Trans. Graph. 34 (4), http://dx.doi.org/10.1145/2766978.

Cho, J., Yun, S., Lee, K., Choi, J.Y., 2017. Palettenet: Image recolorization with given color palette. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 1058–1066.

Colordot, 2020. Colordot - a color picker for humans. URL: https://color.hailpixel.com/. (Online; Accessed 23 June 2020).

Colormind, 2020. Colormind - the AI powered color palette generator. URL: http://colormind.io/. (Online; Accessed 23 June 2020).

COLOURlovers, 2020. COLOURlovers: Color Trends + Palettes. URL: https://www.colourlovers.com. (Online; Accessed 23 June 2020).

Coolors, 2020. Coolors - the super fast color schemes generator! URL: https://coolors.co/. (Online; Accessed 23 June 2020).

da Silva, R.R.O., Paiva, J.G.S., Telles, G.P., Zampieri, C.E.A., Rolli, F.P., Minghim, R., 2019. The visual SuperTree: Similarity-based multi-scale visualization. Vis. Comput. 35 (6), 1067–1080. http://dx.doi.org/10.1007/s00371-019-01696-5.

Jalal, G., Maudet, N., Mackay, W.E., 2015. Color portraits: From color picking to interacting with color. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. CHI '15, Association for Computing Machinery, New York, NY, USA, pp. 4207–4216. http://dx.doi.org/10.1145/2702123.2702173.

Jeong, T., Yang, M., Shin, H.J., 2019. Succinct palette and color model generation and manipulation using hierarchical representation. Comput. Graph. Forum 38 (7), 1–10. http://dx.doi.org/10.1111/cgf.13811, URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13811.

Kita, N., Miyata, K., 2016. Aesthetic rating and color suggestion for color palettes. Comput. Graph. Forum 35 (7), 127–136. http://dx.doi.org/10.1111/cgf.13010, URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13010.

Lamping, J., Rao, R., 1996. The hyperbolic browser: A focus+ context technique for visualizing large hierarchies. J. Vis. Lang. Comput. 7 (1), 33–55.

Lin, S., Hanrahan, P., 2013. Modeling how people extract color themes from images. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '13, Association for Computing Machinery, New York, NY, USA, pp. 3101–3110. http://dx.doi.org/10.1145/2470654.2466424.

Meier, B.J., Spalter, A.M., Karelitz, D.B., 2004. Interactive color palette tools. IEEE Comput. Graph. Appl. 24 (3), 64–72.

Mellado, N., Vanderhaeghe, D., Hoarau, C., Christophe, S., Brédif, M., Barthe, L., 2017. Constrained palette-space exploration. ACM Trans. Graph. 36 (4), http://dx.doi.org/10.1145/3072959.3073650.

Mozilla, 2020. Mozilla web docs - color picker tool. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool. (Online; Accessed 23 June 2020).

O'Donovan, P., Agarwala, A., Hertzmann, A., 2011. Color compatibility from large datasets. In: ACM SIGGRAPH 2011 Papers. SIGGRAPH '11, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/1964921.1964958.

Orchard, M.T., Bouman, C.A., 1991. Color quantization of images. IEEE Trans. Signal Process. 39 (12), 2677–2690.

Paletton, 2020. Paletton - The color scheme designer. URL: https://paletton.com/. (Online; Accessed 23 June 2020).

Phan, H.Q., Fu, H., Chan, A.B., 2018. Color orchestra: Ordering color palettes for interpolation and prediction. IEEE Trans. Vis. Comput. Graphics 24 (6), 1942–1955.

Reingold, E.M., Tilford, J.S., 1981. Tidier drawings of trees. IEEE Trans. Softw. Eng. (2), 223–228.

Schulz, H.-J., Hadlak, S., Schumann, H., 2010. The design space of implicit hierarchy visualization: A survey. IEEE Trans. Vis. Comput. Graphics 17 (4), 393–411.

Schulz, H.-J., Schumann, H., 2006. Visualizing graphs- A generalized view. In: Tenth International Conference on Information Visualisation. IV'06, IEEE, pp. 166–173.

Shackel, B., 2009. Usability–context, framework, definition, design and evaluation. Interact. Comput. 21 (5–6), 339–346.

Shugrina, M., Lu, J., Diverdi, S., 2017. Playful palette: An interactive parametric color mixer for artists. ACM Trans. Graph. 36 (4), http://dx.doi.org/10.1145/3072959.3073690.

Shugrina, M., Zhang, W., Chevalier, F., Fidler, S., Singh, K., 2019. Color builder: A direct manipulation interface for versatile color theme authoring. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. CHI '19, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/3290605.3300686.

Tennekes, M., de Jonge, E., 2014. Tree colors: Color schemes for tree-structured data. IEEE Trans. Vis. Comput. Graphics 20 (12), 2072–2081.

Xiao, C., Han, C., Zhang, Z., Qin, J., Wong, T.-T., Han, G., He, S., 2020. Example-based colourization via dense encoding pyramids. Comput. Graph. Forum 39 (1), 20–33. http://dx.doi.org/10.1111/cgf.13659, URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13659.

Zhang, Q., Xiao, C., Sun, H., Tang, F., 2017. Palette-based image recoloring using color decomposition optimization. IEEE Trans. Image Process. 26 (4), 1952–1964.

Raja Mubashar Karim received a B.S. degree in Industrial Design from Korea Advanced Institute of Science & Technology, KAIST, in 2013. He joined the graduate school of Life Media interdisciplinary Program at Ajou University, Suwon, Korea, in 2013 for Integrated Masters with a Ph.D. (iPhD) in Life Media Interdisciplinary Program. He worked as a student researcher at Integrated Design Laboratory, Department of Digital Media, Ajou University, under Professor Kyungwon Lee's guidance. In 2020, he received a Ph.D. degree in Media Contents later continued in the same lab as a postdoctoral fellow with Professor Kyungwon Lee. He is interested in User Experience Design, Human–Computer Interaction, Information Visualization, Visual Analytics.

Taehong Jeong received a B.Sc. degree in digital media from Ajou University, Korea, in 2014. From the same university, in 2021, he also received a Ph.D. degree in media contents. He worked as a student researcher at Visual Computing Laboratory, Department of Digital Media, Ajou University, under the guidance of Prof. Hyun Joon Shin. He is interested in applying computational color technologies in the visual computing application area. Specifically, his research work covers image color manipulation, wide color gamut imaging, and augmented reality.

Hyoji Ha received a B.S. degrees in Digital Media from Ajou University, Korea, in 2013. And progress on Integrated Masters with a PhD (iPhD) in Life media Interdisciplinary Program, at Ajou University, Suwon, Korea, in 2013. He is joined the graduate school of Lifemedia interdisciplinary program at Ajou University, Suwon, Korea, in 2013. He is currently Ph.D course in the Life media Interdisciplinary Program, Ajou University. He is interested in Information Visualization, Visual Analytics, Visual Metaphor, Human-Computer Interaction and User Experience Design.

Jaejong Ho received a B.S. degrees in Digital Media from Ajou University, Korea, in 2019. He joined the graduate school of Lifemedia interdisciplinary program at Ajou University, Suwon, Korea, in 2019. He is currently master course in the Life media Interdisciplinary Program, Ajou University. He is interested in information visualization.

Kyungwon Lee received a MFA degree in computer graphics and interactive media from the Pratt Institute, USA, in 2002. He joined the faculty of the Department of Digital Media at Ajou University, Suwon, Korea, in 2003. He is currently a Professor in the Department of Digital Media at Ajou University. His research interests include Information Visualization, Human-Computer Interaction and Media Art.

Hyun Joon Shin received the BS, MS and PhD degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST) in 1995, 1997 and 2002, respectively. Since 2005, he has been with the Department of Digital Media at Ajou University, where he is currently a professor, teaching computer graphics and computer vision. His recent research interests include computer graphics, image processing, and computational photography.