

# Playful Palette: An Interactive Parametric Color Mixer for Artists

MARIA SHUGRINA, Adobe Research and University of Toronto

JINGWAN LU, Adobe Research

STEPHEN DIVERDI, Adobe Research

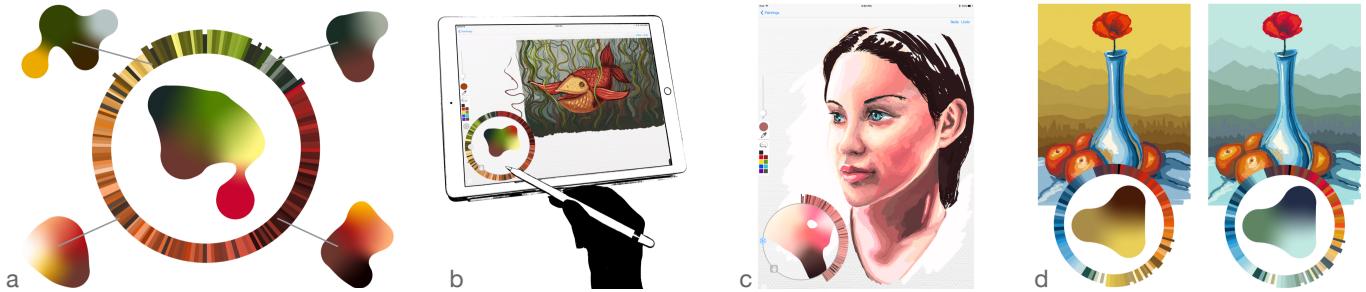


Fig. 1. Our parametric palette representation (a), enables an interactive palette interface for digital painting (b), which allows an artist to explore and access custom color gamuts tailored to their work (c). The artist can seamlessly switch between mixing dishes in the infinite palette history by touching the color history wheel (a), and reverse past color choices in palette history, automatically repainting the artwork (d). Selecting a color from the history wheel recalls the palette configuration at the time the color was used (insets on left).

We present Playful Palette, a color picker interface for digital paint programs that derives intuition from oil paint and watercolor palettes, but extends them with digital features. A Playful Palette is a set of blobs of color that blend together to create gradients and gamuts. They can be directly manipulated to explore arrangements and harmonies. All edits are non-destructive, and an infinite history allows previous palettes to be revisited and modified, recoloring the painting. The Playful Palette design is motivated by a pilot study of how artists use paint palettes, and we evaluate the final design with a set of traditional and digital media painters to demonstrate that Playful Palette is effective both at enabling artists' color tasks, and at amplifying their creativity.

CCS Concepts: • Human-centered computing → Graphical user interfaces; Gestural input; • Applied computing → Media arts;

Additional Key Words and Phrases: creativity support, digital painting, color, parametric color gamut representation, interactive interfaces

## ACM Reference format:

Maria Shugrina, Jingwan Lu, and Stephen DiVerdi. 2017. Playful Palette: An Interactive Parametric Color Mixer for Artists. *ACM Trans. Graph.* 36, 4, Article 61 (July 2017), 10 pages.

DOI: <http://dx.doi.org/10.1145/3072959.3073690>

## 1 INTRODUCTION

An important part of the digital painting process that has not received enough research attention is the design of the color picker. Most software relies on old interface concepts built around selecting RGB or HSV slider values directly or choosing a color from a palette

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2017 ACM. 0730-0301/2017/7-ART61 \$15.00

DOI: <http://dx.doi.org/10.1145/3072959.3073690>

of swatches. This is unfortunate because choosing and composing colors is a critical part of any painting process. We conducted a pilot study that found artists interact with their palette several times a minute, and many of the interactions (such as exploring harmonizing colors) are not well-supported by digital color pickers. Some software [Adobe 2011; Corel 2017] mimics a real paint palette to allow mixing of colors, but such a literal adaptation of the physical interface loses many of the advantages of digital tools such as non-destructive editing.

We present Playful Palette, a new color picker interface that is designed to support the different tasks artists use paint palettes for, while employing an easy interaction mechanism that allows rapid exploration and creative inspiration. A Playful Palette consists of a number of color blobs that blend with nearby blobs within some distance threshold (Fig. 1a), allowing the user to quickly select mixed colors along the resulting gradients. Blobs can easily be moved or recolored to create new color combinations and new gradients, while an infinite history ensures no previous combination is ever lost. Multiple palettes can be created and accessed using history to represent different parts of the painting, such as a palette of blues, purples, and white for the sky, or peach, olive, beige, and brown for skin tones.

The primary goal of Playful Palette is to directly support the color selection needs of digital painters. The secondary goal is to make artists more creative, which we support in two ways. First, the non-destructive editing and spatial arrangement capabilities make it easy for artists to experiment with different harmonies and gradients, which lets them evaluate different palettes more rapidly and effectively. Second, following research indicating that positive emotion boosts creativity [Lewis et al. 2011], Playful Palette incorporates elements of gamification in its interactive, direct manipulation of dynamic blobs.

Playful Palette enables other novel editing operations. Because of the infinite history tracking, after a color from the palette has been used to paint part of the canvas, that palette can be retrieved and the palette blob's color can be changed, causing the painting to be re-colored accordingly (Fig. 1d). This operation is similar to photo recoloring work [Chang et al. 2015] but uses the semantic mapping between Playful Palette and the painting. This allows the artist to continue exploring his or her chosen color gamut even after spending a considerable time painting.

The contributions of this work are as follows. First, we conduct a pilot study with artists to understand how they use and think about color and report insights useful for future color picker interface designs. Then, we propose the parametric representation and interface design of Playful Palette. Next, we detail how infinite history is used for painting recoloring. Finally, we conduct a user study with digital and traditional media artists to evaluate how effectively Playful Palette enables artists and improves their creativity.

## 2 RELATED WORK

Many digital color representations have been developed beyond RGB, including more perceptually uniform color spaces such as CIELAB [CIE 2001], spaces with more intuitive dimensions such as HSV [Smith 1978] and HWB [Smith and Lyons 1996], as well as color spaces most representative of human color perception [Derington et al. 1984; MacLeod and Boynton 1979]. Early studies compare the effect of the color space on user performance in the color picking task [Schwarz et al. 1987], with later work showing that interface design has a more significant influence on usability than color space [Douglas and Kirkpatrick 1999]. The focus of our work is specifically the design for a color picker interface.

Previous research has addressed the issue of color picker interface design as early as Bauersfeld and Slater [1991], who examine the importance of selecting colors in context and exploring variants. Meier et al. [2004] propose a number of interfaces for exploring spatial arrangement of colors, as well as a gradient mixer interface allowing a user to set up a graph of pairwise gradients, and more. While their work explores a rich set of ideas on working with color, they do not propose a single consolidated user experience addressing a specific set of needs, but rather a disjoint collection of widgets, and provide no qualitative or quantitative evaluation. In contrast, we perform a field study to define a clear set of artist needs in the context of digital painting and propose and evaluate a single practical interface addressing all of these needs. Similarly to our work, Jalal et al. [2015] perform a field study and consolidate a set of common practices for working with color in a Color Portraits design space. While their study is broader than ours in context, with the domain spanning any creative task and tool, they only offer a set of “probe” interfaces with limited explanation and evaluation. The more focused nature of our study allows us to provide quantitative measures of a palette’s uses in the context of painting and to deliver a consolidated user interface solution.

A number of digital painting systems have interfaces that simulate real paint palettes [Adobe 2011; Baxter et al. 2001; Blatner et al. 2011; Corel 2017; Vandoren et al. 2008]. While these tools are effective at reproducing the desirable behaviors of the physical painting process, they also come with its physical limitations. These

limitations include destructive editing, inability to restore palette to the previous state, inability to quickly reorganize the colors, and inability to edit the colors after mixing. Digital paint systems have also used physically based pigment models to achieve greater realism in color mixing. Curtis et al. [Curtis et al. 1997] use a 3-wavelength version of the Kubelka-Munk equations and select physical pigments via a regular color picker. The IMPaSTo system [Baxter et al. 2004] approximates oil paint pigments with a reduced dimensional model, letting users select discrete paint colors and then mix them. RealPigment [Lu et al. 2014] estimates composited colors from photos of real color charts, with regular RGB picking. Playful Palette takes inspiration from artists’ interaction with real paints while attempting to avoid their limitations. Our method is not limited to a specific color space, and we explore physical pigments in addition to RGB color mixing.

There has been considerable research in the related problem of choosing good color palettes automatically, for painting and drawing but also for graphic design, photography, and visualization. There is work on using data to infer color compatibility [O’Donovan et al. 2011, 2014], and many methods on constructing color palettes automatically from images [Chang et al. 2015; Jahanian et al. 2015; Lacroix 2010; Lin and Hanrahan 2013; Nguyen et al. 2015]. These results are orthogonal to the problem of designing a user interface to help artists create and organize color gamuts, as automatic methods do not replace such tools. Palette selection has also been addressed for specific applications. For graphic design, there is work to assign colors from a palette to a pattern [Lin et al. 2013], to expand the palette of an artwork with additional colors [Kita and Miyata 2016], and to suggest harmonious colors to an artist while they paint [Son et al. 2015]. Other researchers have developed specific interfaces to support color palette selection for functional tasks including cartography [Harrower and Brewer 2003], scientific visualization [Mueller et al. 2008], and information visualization [Gramazio et al. 2017]. These works all sacrifice elements of artistic control to achieve convenience, which is at odds with the desires of professional artists. Conversely, Playful Palette provides full flexibility and does not attempt to aid the artist’s aesthetic choices.

## 3 ARTIST NEEDS

Part of painting with traditional media is learning effective palette layout, as well as strategies for mixing desired colors from a set of well-known pigments. It is unclear what specific importance the standard painter’s palette has in the creative process beyond simply fulfilling the mechanical needs of mixing new colors and loading the brush. If there are other functions of the painter’s palette, then they should be reproduced in a digital color picker interface. To gain greater understanding of this issue, we conducted a **pilot user study** with practicing artists. Our findings suggest that the palette is an essential part of the creative process and that a number of its functions are not well supported in digital painting tools. From this insight we formulate a set of requirements for a digital color picker which informs the design of the Playful Palette interface.

### 3.1 Pilot User Study

We conducted the pilot study with 12 artists, comprised of 4 professional traditionally trained artists, 4 professional digital artists,

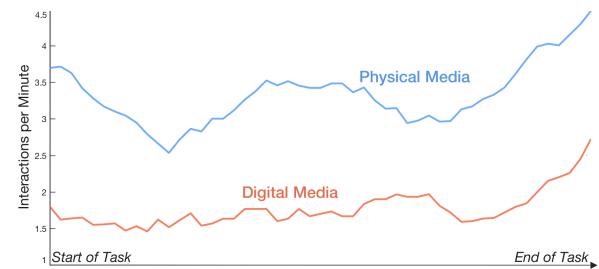


Fig. 2. OS X's system color picker options, which are representative of most digital paint program color pickers. They allow users to select colors using a wheel, sliders, or swatches.

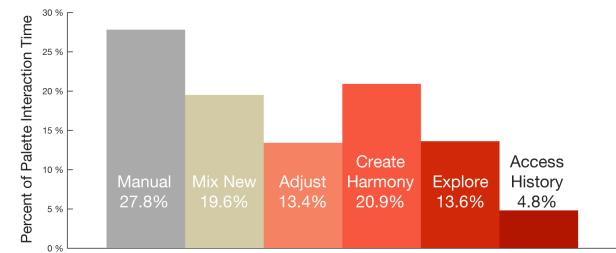
and 4 amateurs with an interest in art. We asked each participant to paint an image using their media of choice (**Task 1**), and then paint the same image using iPad Pro and Apple Pencil with Adobe Sketch [2016b] (**Task 2**), which features a standard HSV wheel color picker (Fig. 2) and records a history of colors used. All but one participant chose physical media for Task 1, selecting one of oil, acrylic or watercolor paints. Approximately 15 minutes were allocated for each painting. We asked the artists to verbalize their process for working with color and point out shortcomings and advantages of either interface, and recorded videos of their work. All recorded palette interactions were manually coded and categorized.

Working with the palette or color picker was the second most common user interaction after painting. On average, users interacted with the physical palette 3.2 times per minute and with the digital color picker 1.7 times per minute, with interaction frequency fluctuating, but remaining consistently high throughout both tasks (Fig. 3a). We further break down time spent interacting with the physical media palette to understand its specific uses (Fig. 3b). Users spent only 28% of time fulfilling a manual need of the medium, such as reloading or cleaning a brush, and 33% mixing a new color or adjusting the current color. The remaining 40% served functions not currently supported by most digital color interfaces, such as creating harmonizing color combinations by tempering colors with others that already occur in the painting, exploring different color combinations and variations, and accessing color history, e.g. returning to a section of the palette to rework a part of the painting. Note that while accessing historical colors takes only 4.8% of the time, 10% of user interactions fell into that category.

The verbal feedback during our study was also informative. Some of our participants described mixing colors as “therapeutic,” “tactile,” and “organic.” It is clear that many artists enjoy color mixing, as one called a pigment “a really fun color” and another said she may “spend more time [mixing] than [painting].” The serendipity of real paints was made clear by one artist’s comment that “you never really know exactly the color you want until you arrive at it by mixing;” another said, “it seems like there are only two colors, and then you start to play with them and refine more and more;” and a third said while mixing, “I’m not quite attempting to get this color...I’m really just getting an intuition.” Finally, it was clear that some of



(a) Number of palette/color picker interactions per minute using physical and digital media throughout the duration of the painting task, averaged over 11 users with a 3 minute averaging window.



(b) Average distribution of the time users spent interacting with the palette when using physical media, broken down into 6 categories. Average distribution across 11 users.

Fig. 3. **Pilot User Study Findings:** a) Prevalence of palette/color picker interactions. b) A large fraction of physical palette interactions fall into the categories poorly supported by the digital color pickers.

our participants did not care for digital color pickers, from their comments such as “It’s less enjoyable here. Too many parameters here. I don’t need all this yellow,” and “it’s such a digital way to think about paint, like RGB.”

Independently of our pilot study, we found similar sentiments from other artists: “I love colour theory but nothing makes me happier than knowing cadmium orange and cerulean blue make a truly beautiful shade of grey. Mixing these two pigments together can be temperamental. They don’t always turn out. But when they do? Pure magic.” [Molčan 2016] Clearly for many artists, color is more about an emotional response, rather than numeric precision.

### 3.2 Requirements for a Palette Interface

Based on our pilot study findings, we have distilled a set of requirements for a palette interface. To fully support the needs of artists, a color palette must include all of these capabilities:

- R1 **Select:** efficiently create intended color
- R2 **Blend**<sup>1</sup>: create compatible color by blending colors
- R3 **Explore:** rapidly try different color combinations
- R4 **History:** easily return to color already used
- R5 **Organize:** map out the color gamut for an artwork

<sup>1</sup>We observed artists creating harmonious color combinations by combining colors on the physical palette, and so focus on blending in the definition. It is possible that harmonization can be fulfilled by generating compatible colors through other means.



Fig. 4. Fulfillment of our palette interface requirements by various interfaces from commercial software and previous work.

In effect, this set of requirements provides a more detailed characterization of ‘Palette’ and ‘History’ practices from the more general Color Portraits design space [Jalal et al. 2015] in the context of painting. Because digital color pickers (Fig. 2) already provide support for R1 and are a standard familiar feature of digital interfaces, we design our palette interface to work in conjunction with these tools, and augment their R1 capability. Playful Palette fulfills R1-R5 in a single cohesive interface, unlike other color picker interfaces.

### 3.3 Comparison of Interfaces

We evaluate existing color picker interfaces within the framework of our palette requirements (see summary in Fig. 4).

**Standard Digital Tools:** Digital color pickers (Fig. 2) are well-tuned to support R1 by providing many different mechanisms to adjust a color, via RGB or HSV sliders, other color spaces and even numeric entry. While it may appear that these interfaces also support exploration (R3), they are severely limited by allowing exploration of only one color at a time. In addition, most digital interfaces are equipped with an eyedropper tool that allows sampling colors from the canvas, and so supports history (R4), but only in a limited sense, as the history of arriving at the canvas color by blending, compositing or other means is lost. R4 is also partially supported by swatch palettes (Fig. 2 right), which allow an artist to manually save important colors. Some interfaces [Adobe 2016b] do keep track of color history, but it is typically hidden in a separate widget, and not easily accessible. Blend (R2) and organize (R5) requirements are missing completely. Some artists employ workarounds, such as using a swatch palette to organize a color gamut, or using a section of the canvas as a palette, but because the interfaces are not designed for such uses, they are brittle and cumbersome.



Fig. 5. Pilot Study: Example artwork produced by the same artist using physical media (A1) and digital media (A2), and the corresponding palette (P1) and digital color history (P2).

**Physical Palette:** A physical paint palette (Fig. 5) on the other hand more directly satisfies the opposite requirements. A strength of physical palettes is exploration (R3), as color combinations can yield surprising results adding an element of serendipity to the process, but conversely making R1, selecting a specific color, much harder. Palettes afford R5, organization, by enabling artists to arrange their chosen colors around the physical space as they please. This spatial arrangement presents the whole color gamut simultaneously, which then allows harmonization through blending, R2, by adding a new color, e.g. a dab of red, to all the colors on the palette. However, once set, the palette cannot be rearranged, causing it to become messy over time. Physical palettes have all the drawbacks of physical media, such as the lack of any sort of undo or history capability, which limits how well they can support R4, accessing color history—once a mixed paint color runs out, it can be very difficult to reproduce. Non-destructive editing is also important for support of R2, R3, and R5, as they all entail experimentation and the inability to undo means that mistakes can have irrecoverable consequences. This may cause artists to be more conservative with their exploration than they would be otherwise. For these reasons, simply emulating the physical palette with a digital version is not the best design to support artists’ color needs.

**Blending Interfaces:** The iPad app Paper [2016] includes a novel color picker that allows two colors to be gradually mixed according to a data-driven algorithm learned from aesthetically pleasing example gradients [Dannen 2012]. However, blending does not extend beyond 2 colors, limiting blending and exploration to a single curve in RGB space, and R4 and R5 are as poorly supported as with standard digital interfaces. Meier et al.’s [2004] color mixer interface similarly supports color gradients only between pairs of colors, limiting R2 and R3, and, crucially, provides no history, R4. Paint-like mixing interfaces emulating the physical palette [Adobe 2011; Baxter et al. 2001; Blatner et al. 2011; Corel 2017; Vandoren et al. 2008] suffer from the same limitations as the real physical palettes.

**Color Theme Interfaces:** Many projects have developed interfaces for creating aesthetic color palettes. Adobe Color CC [2016a] is an interface that helps users pick colors that harmonize according to traditional design rules. Hu et al. [2014] present an interface that selects colors based on a rhythmic principle, while Wijffelaars et al. [2008] use a small number of controls to parameterize

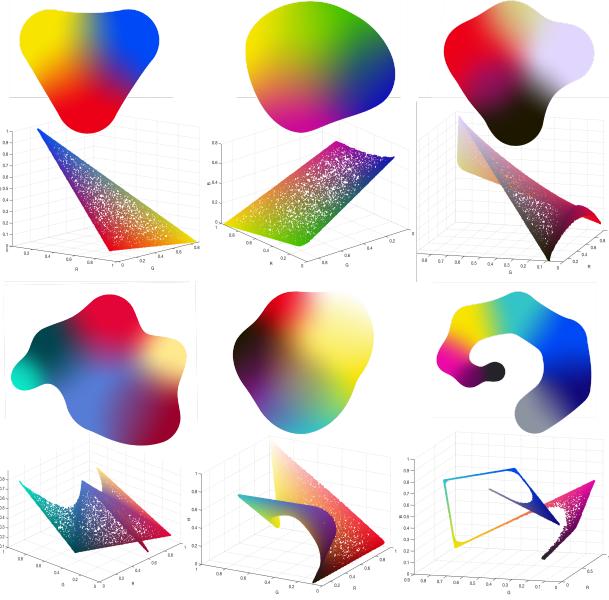


Fig. 6. Left to right, top to bottom: mixing dishes with 3, 4, 5, 6, 7 and 8 blobs, respectively, and the corresponding color distributions sampled from the palettes and plotted in sRGB space.

pleasing curves through perceptual color space. Finally, Heer and Stone [2012] map semantic names to colors and use this analysis to choose better palettes by understanding color salience and ambiguity. While these approaches support exploration, R3, they work with small numbers of discrete colors, which is insufficient for a number of creative tasks such as digital painting and illustration. They are also generally missing support for the other requirements. We support palettes for more complex continuous color gamuts and cover the other requirements in a single interface.

**Playful Palette:** Rather than attempting to replace what standard digital tools already do well, Playful Palette augments their existing capabilities by additionally fulfilling blending (R2), exploring (R3), and organizing (R5) through interactive color gamut construction. We enhance selection (R1) by allowing the artist to select colors from the resulting continuous gamut. Our infinite palette history provides improved color recall (R4), and its application to painting recoloring further enables exploration (R3).

#### 4 PLAYFUL PALETTE DESIGN

We propose a simple parametric palette representation that is easy and intuitive to edit, fast to render and very compact to store. The representation is generalizable to other digital color operations and physical color models, and supports full history tracking. Our design meets all our requirements while avoiding many shortcomings of the traditional palette.

##### 4.1 Parametric Palette Representation

We represent a palette  $P$  for a given artwork as a set of mixing dishes  $D_1 \dots D_n$ , where each mixing dish is a set of color blobs  $B_i$ . Each blob  $B_i$  is a tuple of attributes  $\{c_i, p_i, r_i, m_i\}$ , containing color

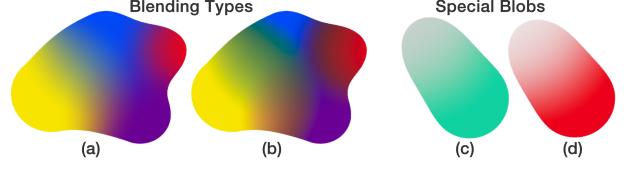


Fig. 7. Different blending types using our palette representation: a) sRGB interpolation, b) interpolation in the spectral coefficient space of real paints rendered using the Kubelka-Munk equation, c) d) desaturation blob in HSV space blended with turquoise and red blobs.

vector  $c_i$  in a chosen color space  $\mathbb{K}$  (e.g. sRGB), position  $p_i$  in  $\mathbb{A}^2$ , a continuous subset of  $\mathbb{R}^2$  (e.g. we use the area defined by the unit circle), radius  $r_i$ , and an optional metadata vector  $m_i$ . We define a family of continuous, smooth functions  $\mathcal{F}(p, r) \rightarrow (\mathbb{A}^2 \rightarrow \mathbb{R}_+)$ , where each function defines a local area of influence of each blob. The position and radius parameters of each blob  $B_i$  implicitly define its influence function  $M_i := \mathcal{F}(p_i, r_i)$  over the domain  $\mathbb{A}^2$ . We use a variant of the metaball function [Blinn 1982; Wyvill and Wyvill 1989], which is a Gaussian approximation with finite extent:

$$\mathcal{F}(p_i, r_i)(p) = \begin{cases} 1 - \frac{4d^6}{9b^6} + \frac{17d^4}{9b^4} - \frac{22d^2}{9b^2} & d \leq b \\ 0 & \text{else} \end{cases}$$

where  $b$  is a constant, and  $d$  is the distance from point  $p$  to the center  $p_i$  of blob  $B_i$ . In practice, we transform  $d$  so the falloff from a blob of any radius is the same as for a blob of some ideal radius  $\hat{r}$ . We set  $b = \hat{r} + m$ , where  $m$  is the falloff margin around the blob, with  $\hat{r} = 0.2$  and  $m = 0.3$ .

To render a palette dish  $D_j$ , we render all the points  $p \in \mathbb{A}^2$  satisfying:

$$\sum_{B_i \in D_j} M_i(p) \geq T \quad (1)$$

For some threshold  $T > 0$ , this results in a smooth filled-in isosurface of the blobs in the palette. To compute the color  $c_p \in \mathbb{K}$  at  $p$ , we use the blob influences  $M_i(p)$  as interpolation weights:

$$c_p = \frac{\sum_{B_i \in D_j} M_i(p) \cdot c_i}{\sum_{B_i \in D_j} M_i(p)} \quad (2)$$

Eq. 1 and Eq. 2 can be trivially computed in parallel in a fragment shader, as in our prototype.

This parametric palette representation is simple to implement, easy to edit interactively, and can parameterize a wide variety of color manifolds (Fig. 6), which makes this representation a powerful and attractive mechanism for constructing custom color spaces and exploring color gamuts.

While our standard model uses sRGB colors (Fig. 7.a and all other figures), it can also interpolate in other color spaces. For example, instead of a 3D sRGB vector,  $c_i$  can be a vector of physical Kubelka-Munk (KM) coefficients [Hecht 1983]. The coefficients are still interpolated by Eq. 2 and the color can be rendered with the KM equation. We implement this with a set of real acrylic paint coefficients [Okumura 2005] as an alternative to sRGB (Fig. 7.b). It is also possible to implement special “transformer” blobs such as desaturators, which control only the S component of the HSV color space (Fig. 7 c, d).

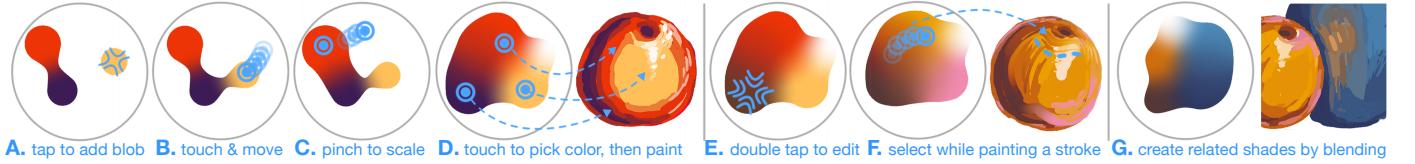


Fig. 8. **Playful Palette interactions:** User creates a palette by changing color with HSV color picker and tapping in the palette dish to add a color blob, moves blobs around by touching and moving, resizes them by touching and pinching in or out.

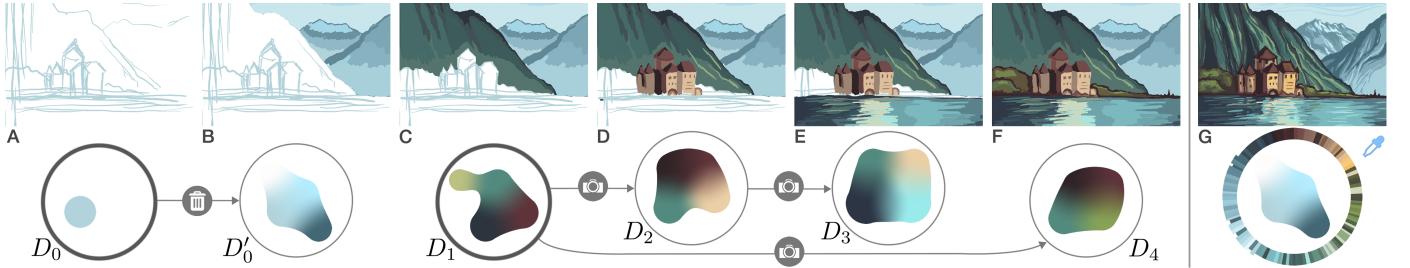


Fig. 9. **Palette History:** During painting (A-F), the user creates new palette dishes from scratch ( $D_0$ ,  $D_1$ ) and edits existing dishes. If the edited palette (e.g.  $D_2$ ) is sufficiently different from its parent ( $D_1$ ), a new edited snapshot is automatically created. Otherwise, the edited dish replaces the original (e.g.  $D'_0$  replaces  $D_0$ ). A palette dish can be retrieved from history by re-selecting a color that has been mixed with it from the color history wheel or by picking a color from the painting (G).

## 4.2 Interactive Editing

Our palette representation enables a responsive and expressive palette interface, which we implement on an iPad Pro using multi-touch gestures<sup>2</sup>. In our prototype, a user can create a new mixing dish, add, remove and rearrange color blobs, and change their color. Our prototype is designed to fulfill our palette requirements. Because the standard HSV color picker is well suited for selecting a color (R1), we incorporate it to work in conjunction with the Playful Palette to fulfill R2-R5.

Fig. 8 shows an example Playful Palette interaction. Suppose an artist is painting a still life with a vase and fruit. Even if the scene is set before him or her, painting involves more than replicating colors. The fruit could be saturated with dramatic violet shadows and stark highlights (Fig. 8D), or more realistic, with brown shadows and subtle hues of pink (Fig. 8F). Such choices influence the style and mood, which Playful Palette helps the artist explore (R3) by adding blobs, rearranging, and scaling them (Fig. 8ABC). The blobs are moved via touching and dragging, creating new gradients with a smooth, fluid-like behavior. The artist can touch the palette to select a mixed color and paint (Fig. 8D). Double tapping a blob opens an HSV picker to change its color and thus the gamut (Fig. 8E), to paint another fruit. Once satisfied with the color arrangement (R5), the palette can be recalled for painting more fruit (Fig. 8F). When painting a reflection of the fruit in the vase, the artist can copy and paste a color from the fruit palette to the vase palette (Fig. 8G), blending colors to create a harmonious gradient (R2).

## 5 HISTORY AND RECOLORING

Our prototype stores the *color history* of a painting, similar to applications such as Adobe Sketch [2016b], as the set of colors

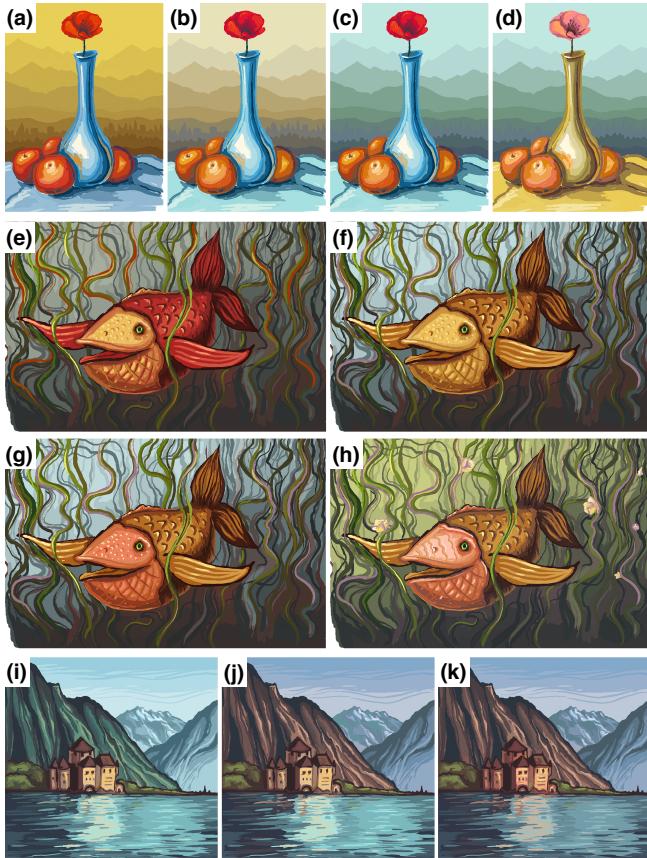
<sup>2</sup>With minor changes, these interactions could be adapted for a mouse interface.

$H^c = \{c_1 \dots c_k\}$  that were used to paint at least one stroke. Unlike other digital palette interfaces, Playful Palette stores the *palette history* itself. This is made possible by our compact palette representation. Storing the palette history of simulated physical media [Baxter et al. 2001] would require saving per-pixel pigment data. In contrast, each blob in a Playful Palette takes 8 bytes and a typical mixing dish contains a small number of blobs.

### 5.1 Palette History

The palette history is equivalent to the palette  $P$  itself, where each mixing dish  $D_i$  can be explicitly created by the user or can be saved automatically during palette editing. Each dish  $D_i$  references its parent mixing dish  $D_i^P$ , if any, as well as a list of colors  $H_i^c \in H^c$  that were picked from  $D_i$ . A snapshot of the dish state is only saved if the new dish no longer represents  $H_i^c$ . For example, in Fig. 9 the user edits  $D_0$  to produce  $D'_0$ , and the previous palette  $D_0$  is discarded because  $D'_0$  still represents the same colors. Then the user creates a mixing dish  $D_1$  and later edits it to produce  $D_2$ , which no longer has the light green shades. This causes  $D_1$  to be saved automatically as a separate snapshot.

To visualize palette  $P$ , we only display one mixing dish  $D_i$  at a time, and arrange  $H^c$  by hue around its perimeter (Fig. 9G). Touching a perimeter color  $c_j$  will populate the palette with  $D_f$  s.t.  $c_j \in H_f^c$ . Similarly,  $D_f$  can be recalled by selecting any element of  $H_f^c$  from the canvas using a standard eyedropper tool. As it is possible to revisit a dish  $D_i$  and edit it, the palette history is a forest of trees, where each dish has at most one parent (Fig. 9). Since a color may be represented on more than one mixing dish, selecting a dish by a color on the canvas can be ambiguous. Therefore, every canvas pixel should be tagged with the mixing dish ID used to paint it.



**Fig. 10. Painting Recoloring:** Results of painting recoloring by editing the palette history and propagating the color changes to the strokes in the image. Independent mixing dishes have independent effects on the colors (top row: flower and oranges can be recolored independently). Repainting can be interleaved with painting (fish example, recoloring is followed by more painting and recoloring). Related palettes affect colors together (in the castle example, changing the castle color affects its reflection, because these palettes are linked in the history.)

## 5.2 Painting Recoloring

We take advantage of the association between the palette history and the canvas colors to enable painting recoloring that is history-aware. In **painting recoloring mode** only changes to blob colors are allowed (they may not be moved, added, or deleted). When a mixing dish  $D_i$  is edited by changing the color of blob  $B_c$ , the edit is propagated to all the descendant dishes  $D_1^i \dots D_k^i$ . If  $B'_c$  in a child dish  $D_j^i$  is the same color as its parent, then it is updated to the new color and  $D_j^i$  becomes  $\tilde{D}_j^i$ . For example in Fig. 9, changing the color of the brown blob to color  $c$  in  $D_1$  would change the corresponding brown blobs in  $D_2$  and  $D_4$  to  $c$ , but will cause no changes in  $D_3$ .

Once  $\tilde{D}_1^i \dots \tilde{D}_k^i$  are updated, we transfer the edit to the image. Each pixel  $x$  in the painting contains color  $c_x$  and a reference to the dish  $D_x$  used to pick  $c_x$ . We find any location  $p_x$  of color  $c_x$  in the original  $D_x$  before the edit. Then, we update pixel  $x$  with the color at location  $p_x$  in the edited dish  $\tilde{D}'_x$ . Thus, an artist can affect large

areas of the painting by editing palette dishes high up in the history tree or only change the recent details by editing the leaf dishes. After recoloring the artwork, the artist can continue painting with the updated mixing dishes in the palette history.

Examples of recoloring are in Fig. 10. An advantage of our approach is that our palette history can distinguish between edits to similarly colored regions of the painting based on how they were painted, e.g. the water and sky in Fig. 10i, whereas previous work [Chang et al. 2015] would be unable to recolor them separately. This also aids the exploration requirement R3, as the artist is free to explore and change the color gamut even after spending a considerable time painting. A downside of our interface is that it is not always clear what effects changing a palette color will have on an image, particularly if the color was used early on in the painting and so influences many of the descendant mixing dish colors. Our formulation also does not handle opacity effects, and we plan to address this limitation in future work.

## 6 USER STUDY

We implemented a prototype digital painting app with the Playful Palette interface so we could have artists try it out (Fig. 1, 11). More formally, we also conducted a user study to evaluate the interface. As our interface is designed to work in conjunction with the standard HSV color picker, we assess the added benefit of the Playful Palette by comparing a painting task performed with the HSV color picker alone (**HSV task**) to the same task performed with the HSV color picker *and* Playful Palette (**PP task**) in an A/B within-subject user study. Rather than attempting to devise a procedure to evaluate the quality of the produced artwork, which is hard to do objectively, we focus on evaluating our main goal of supporting artists' creativity by directly measuring creativity support. In addition, we report usage statistics and the qualitative experience of our artist participants.

### 6.1 Participants and Methodology

We recruited 14 participants: 8 professional digital artists practicing in fields such as digital painting and graphic design, 4 professional traditional artists with expertise in natural media such as acrylic and oil, and 2 amateur artists practicing painting as hobbies. These users have *not* participated in our pilot user study. Each user was compensated with a \$45 Amazon gift certificate.

We randomly selected the order of HSV and PP tasks, ensuring an even split between the users. Each task was preceded by a 5 minute training session. Following this, we provided the user with a light underdrawing (Fig. 12) and instructed them to paint it using any interpretation and any color scheme, giving them 16 minutes for the task. In addition to logging all user interactions, we administered the Creativity Support Index questionnaire and a custom questionnaire at the end of each task.

### 6.2 Evaluating Creativity Support

The key goal of the Playful Palette is supporting artist's creativity. In order to quantify this effect, we rely on the Creativity Support Index (CSI) [Carroll and Latulipe 2009], which measures creativity support of an interface along six dimensions: Exploration, Expressiveness, Immersion, Enjoyment, Results Worth Effort, and Collaboration (not

relevant to our work). The index is computed based on a well-tested questionnaire, which we administered directly after each task using the implementation provided by Carroll and Latulipe [2009]. Each user answered questionnaire questions about both tasks in the order that they were performed.

CSI Dimension	HSV	PP	T-test
Exploration	26.2	46.3	$t = -4.32, p = 0.00084$
Expressiveness	31.9	42.5	$t = -3.18, p = 0.0073$
Immersion	21.8	31.4	$t = -2.08, p = 0.058$
Enjoyment	20.1	32.9	$t = -2.44, p = 0.030$
Results Worth Effort	26.2	30.6	$t = -0.92, p = 0.37$
CSI	42.1	61.2	$t = -3.40, p = 0.00476$

Table 1. Break down of the CSI scores by dimensions with corresponding paired-samples t-test. We omit Collaboration, which scored 0 on both tasks.

The CSI scores ranging from 0 (worst) to 100 (best) should be used for *comparison* under conditions with few variables, such as for comparing interfaces used by the same population for the same task, and so is well suited to our scenario. The mean CSI score was 42 (std=9.5) for the HSV task, and 61 for the PP task (std=17). A paired-samples t-test shows a significant difference in the CSI scores ( $t=-3.40, p=0.0048$ ), indicating that Playful Palette consistently performs better when quantifying Creativity Support for a painting task. Because the pairs of samples for HSV and PP depended not only on the UI used, but on an additional variable of task order, we tested the effect of both factors using two-way ANOVA. The test did not reject the null hypothesis that the task order has no effect on the CSI score ( $F=0.23, p=0.64$ ), nor does it reject the null hypothesis that there is no interaction between the two factors ( $F=0.06, p=0.82$ ). The test did find a significant effect between the UI design and the score ( $F=12.2, p=0.0019$ ). When broken down into CSI dimensions (Table 1), statistically significant improvement due to Playful Palette was detected for Exploration, Expressiveness and Enjoyment, a finding that is not surprising, given Playful Palette's affordances for expressiveness. This is also supported by the qualitative feedback from the users. We did not find evidence to reject the null hypothesis for Immersion and Results Worth Effort. We hypothesize that these two dimensions are heavily influenced by the usability and other qualities of the painting interface, such as the available brushes, irrelevant to the evaluation of the color picking interface.

### 6.3 Quantitative Metrics

When using the Playful Palette, artists used on average 39% more distinct colors than when using HSV picker alone, with a statistically significant difference (paired-samples t-test:  $t=-2.23, p=0.04413$ ) in the mean number of colors used: 38.2 for the HSV task, and 53.3 for the PP task. On average, users changed color after every 10.2 strokes (std=5.2) in the HSV task, and after every 5.24 strokes (std=1.47) in the PP task. In the HSV task, 74% of the color changes were performed with the HSV picker, and 26% with the eyedropper tool, whereas in the PP task, 75% of color changes were performed by picking color from the Playful Palette mixing dish, and the rest were split between the HSV picker (11%), color history wheel (8%), and the eyedropper tool (7%). In terms of the time users spent painting



Fig. 11. Paintings made by artist (©Daichi Ito) in Playful Palette. The palette for the left painting is shown as inset.

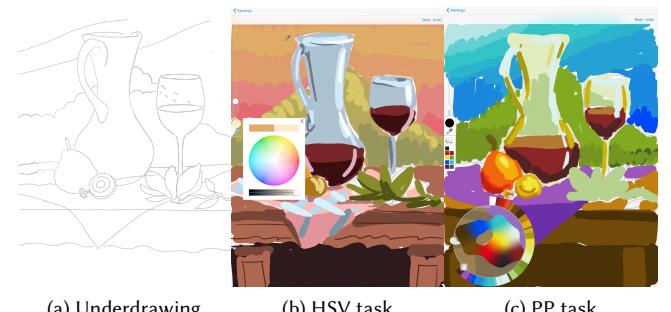


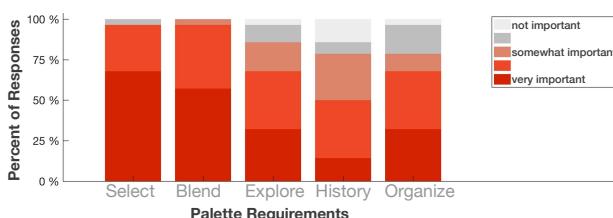
Fig. 12. **User Study:** Users were asked to draw over an underdrawing (a), using HSV picker interface (b), and Playful Palette interface (c), with random order assigned to each user.

and interacting with the palette, users spent 73% of the tracked time (consecutive touches with a gap of no more than 2.5 seconds) painting, 20% interacting with the HSV picker and 6% using the eyedropper tool. With the PP interface, users spent 65% of tracked time painting, 29% interacting with the palette and the remaining 6% using the eyedropper tool or changing the current color using the HSV picker. When working with the palette, users created on average 2.9 independent mixing dishes (not created by editing the current dish). Each mixing dish snapshot contained on average 9.0 blobs (std=3.6). Only 3 users changed blob radius, while all but one changed blob color, on average 9.4 times during the task.

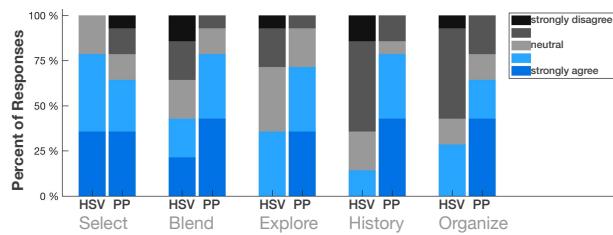
### 6.4 Qualitative Feedback

To validate the conclusions from our pilot study, we asked artists to rate how important our interface requirements are to them when working with color while painting. We asked the same set of questions using a 5-point Likert scale both before and after the study. The responses were similar, and we average them in Fig. 13. Overall, at least 75% of the users found all of the requirements at least somewhat important.

We asked each user to evaluate the performance of the two interfaces, HSV and PP, on all the requirements immediately after they finished the painting task with the corresponding interface. Users generally found the PP interface performed better on explore, blend, history, and organize (R2-R5) (Fig. 13b), and slightly worse on



(a) Artists find the our interface requirements to be important.



(b) Artists reported that Playful Palette (PP) fulfilled R2-R5 better than the HSV-picker interface alone (HSV).

**Fig. 13. Qualitative Feedback:** a) Artist's responses to a questionnaire about importance of our interface requirements. b) Artist's feedback on how well these requirements are fulfilled by using HSV picker alone, or in conjunction with the Playful Palette.

selection (R1), perhaps because Playful Palette invited more exploration at some expense for efficiency. Artists also provided freeform written feedback about the Playful Palette interface. Some of their suggestions included usability improvements for our prototype. For example, some users found it cumbersome to add blobs using a *long press* gesture, and we replaced this with tap in an empty area in the later version of the prototype. Several users requested a larger mixing area, and a more intuitive way to toggle between palettes in the palette history. To this end, we have linked the eyedropper tool to the palette history, so that using the eyedropper on the canvas recalls the palette that was used for that region.

Apart from comments about usability nuances, feedback about the Playful Palette was very positive. Our goal was to create a tool that supports the creative painting process by improving the enjoyment, the expressiveness, and the ease of achieving a state of flow, and therefore the way artists *feel* when using Playful Palette is of paramount importance. Users emphasized how it enabled them to explore and be more creative: “I could easily explore the colors and be creative and I think it would be fun to play with the system a bit more,” “I enjoyed working with the color palette—it felt truer to how I’d go about mixing paint in real life,” “With the palette I felt much more inspired to be creative. I tried to paint as I do when I use acrylics. I explored different ways to use colors and get inspiration from the tool,” and “It’s very relaxing to use, and I’m not afraid to get messy. Color mixing was both very intuitive and also just fun to watch. (I can imagine a lot of kids just dropping a billion colors down to watch them merge. Me too.)”

Users also highlighted particular features in their feedback, such as the history, “I liked how easy it was to change between palette

and go back,” and palette organization, “I liked the ability to create complex gradations of color and to plan out a palette for the piece.” In response to “What do you like about Playful Palette?” one user said, “The ability to re-select previously used colors and modify them slightly. I also liked the way the size of the arc represented the amount of the color used in the painting.”

Finally, many users mentioned enjoying the interface, “I really enjoyed playing with it,” “Very fun to play with. I enjoyed it more and more as I used it,” and “Makes me want to get an iPad so I can paint in bed now.”

## 6.5 User Study Limitations

There are a number of limitations of our study that are important to consider. Because many digital painting tools only include standard tools such as the HSV color picker and eyedropper, we chose this as our baseline condition. We did not compare against recent research interfaces [Jalal et al. 2015; Meier et al. 2004] because none of those interfaces completely satisfy our design requirements (Fig. 4). As Playful Palette is meant to augment a standard color picker, our experimental condition was Playful Palette with an HSV color picker and eyedropper. It is possible this configuration artificially increased our CSI score because a wider variety of tools were available. A future study could design around this, but we felt it was important to test our actual proposed interface configuration.

The effect of novelty could have also influenced our results, as users tend to be more positive about new interfaces. An important long term solution to this problem is to conduct a longitudinal study, but that is outside the scope of our investigation. We attempted to mitigate the effects of novelty by focusing on professional users (as in an expert review study design), who we believe are less susceptible to novelty bias because of their focus on using tools in a work context. It is also possible that professional users have an opposite bias in our study because of their deep familiarity with the HSV picker, which may manifest as an implicit preference and disinclination towards using Playful Palette.

Finally, our study was not blind. The study was administered by the paper authors. This also creates a potential for confirmation bias skewing results towards the experimental condition. On the other hand, because of the pervasiveness of the HSV color picker interface and professional users’ existing familiarity with it, our study participants were already trivially aware of which was the experimental condition, so it is unclear how we could have constructed a double blind methodology.

## 7 CONCLUSION

Based on a pilot study of artists’ needs, we designed Playful Palette, a color picker interface that combines the advantages of both digital and physical painting domains. We introduced a simple and powerful parametric representation for palette that allows non-destructive interactive color editing with infinite color history tracking. We introduced the application of Playful Palette for painting recoloring, which enables an artist to continue exploring color choices well into the painting process. Our user studies confirmed the effectiveness of our interface at achieving the artists’ color goals and amplifying their creativity.

## ACKNOWLEDGEMENTS

We thank the artists who have participated in our user studies for their honest insights and inspiration. We are also grateful to David I.W. Levin for technical discussions, and to Daichi Ito and Brooke Francesi for design critique.

## REFERENCES

- Adobe. 2011. Adobe Color Lava for Photoshop. [https://www.adobe.com/mena\\_en/products/color-lava.html.html](https://www.adobe.com/mena_en/products/color-lava.html.html). (2011).
- Adobe. 2016a. Adobe Color CC. <https://color.adobe.com>. (2016).
- Adobe. 2016b. Adobe Photoshop Sketch. <https://www.adobe.com/products/sketch.html>. (2016).
- Penny F Bauersfeld and Jodi L Slater. 1991. User-oriented color interface design: direct manipulation of color in context. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*. 417–418.
- Bill Baxter, Vincent Scheib, Ming C Lin, and Dinesh Manocha. 2001. DAB: Interactive haptic painting with 3D virtual brushes. In *Proc. Conf. Computer graphics and interactive techniques (SIGGRAPH)*. 461–468.
- William Baxter, Jeremy Wendt, and Ming C Lin. 2004. IMPASTO: a realistic, interactive model for paint. In *Proc. Int. Symp. Non-photorealistic animation and rendering (NPAR)*. 45–148.
- Anthony M Blatner, James A Ferwerda, Benjamin A Darling, and Reynold J Bailey. 2011. TangiPaint: A Tangible Digital Painting System. *Color Imaging Conf.* (2011), 102–107.
- James F Blinn. 1982. A generalization of algebraic surface drawing. *ACM Trans. Graph.* 1, 3 (1982), 235–256.
- Erin A. Carroll and Celine Latulipe. 2009. The Creativity Support Index. In *Extended Abstracts on Human Factors in Computing Systems (CHI EA)*. 4009–4014.
- Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based Photo Recoloring. *ACM Trans. Graph.* 34, 4 (July 2015), 139:1–139:11.
- CIE. 2001. *Improvement to industrial colour-difference evaluation*. Technical Report 142-2001. Central Bureau of the CIE.
- Corel. 2017. Corel Painter. <http://www.painterartist.com>. (2017).
- Cassidy J Curtis, Sean E Anderson, Joshua E Seims, Kurt W Fleischer, and David H Salesin. 1997. Computer-generated watercolor. In *Proc. Conf. Computer graphics and interactive techniques (SIGGRAPH)*. 421–430.
- Chris Dannen. 2012. The Magical Tech Behind Paper For iPad's Color-Mixing Perfection. <https://www.fastcompany.com/3002676/open-company/magical-tech-behind-paper-ipads-color-mixing-perfection>. (2012).
- Andrew M Derrington, John Krauskopf, and Peter Lennie. 1984. Chromatic mechanisms in lateral geniculate nucleus of macaque. *J. Phys.* 357, 1 (1984), 241–265.
- Sarah A Douglas and Arthur E Kirkpatrick. 1999. Model and representation: The effect of visual feedback on human performance in a color picker interface. *ACM Trans. Graph.* 18, 2 (1999), 96–127.
- FiftyThree. 2016. Paper by FiftyThree - Sketch, Diagram, Take Notes. <https://itunes.apple.com/us/app/paper-by-fiftythree-sketch/id506003812?mt=8>. (2016).
- C. C. Gramazio, D. H. Laidlaw, and K. B. Schloss. 2017. Colorgorical: Creating discriminable and preferable color palettes for information visualization. *IEEE Trans. Vis. Comp. Graph.* 23, 1 (Jan. 2017), 521–530.
- Mark Harrower and Cynthia A. Brewer. 2003. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal* 40, 1 (2003), 27–37.
- Harry G. Hecht. 1983. A Comparison of the Kubelka-Munk, Rozenberg, and Pitts-Giovannelli Methods of Analysis of Diffuse Reflectance for Several Model Systems. *Applied Spectroscopy* 37, 4 (1983), 315–403.
- Jeffrey Heer and Maureen Stone. 2012. Color Naming Models for Color Selection, Image Editing and Palette Design. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*. 1007–1016. <http://vis.stanford.edu/papers/color-naming-models>
- Guosheng Hu, Zhigeng Pan, Mingmin Zhang, De Chen, Wenzhen Yang, and Jian Chen. 2014. An interactive method for generating harmonious color schemes. *Color Research & Application* 39, 1 (2014), 70–78.
- A. Jahanian, S. V. N. Vishwanathan, and J. P. Allebach. 2015. Autonomous color theme extraction from images using saliency. In *Imaging and Multimedia Analytics in a Web and Mobile World (Proc. SPIE)*, Vol. 9408.
- Ghita Jalal, Nolwenn Maudet, and Wendy E. Mackay. 2015. Color Portraits: From Color Picking to Interacting with Color. *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2015), 4207–4216.
- N. Kita and K. Miyata. 2016. Aesthetic Rating and Color Suggestion for Color Palettes. *Computer Graphics Forum* 35, 7 (2016), 127–136.
- Vinciane Lacroix. 2010. Automatic Palette Identification of Colored Graphics. In *Proc. Int. Conf. Graphics recognition (GREC)*. 61–68.
- Sheena Lewis, Mira Dontcheva, and Elizabeth Gerber. 2011. Affective Computational Priming and Creativity. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*. 735–744.
- Sharon Lin and Pat Hanrahan. 2013. Modeling how people extract color themes from images. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*. 3101–3110.
- Sharon Lin, Daniel Ritchie, Matthew Fisher, and Pat Hanrahan. 2013. Probabilistic Color-by-numbers: Suggesting Pattern Colorizations Using Factor Graphs. *ACM Trans. Graph.* 32, 4 (July 2013), 37:1–37:12.
- Jingwan Lu, Stephen DiVerdi, Willa A Chen, Connally Barnes, and Adam Finkelstein. 2014. RealPigment: Paint compositing by example. In *Proc. Int. Symp. Non-photorealistic animation and rendering (NPAR)*. 21–30.
- Donald IA MacLeod and Robert M Boynton. 1979. Chromaticity diagram showing cone excitation by stimuli of equal luminance. *J. Opt. Soc. Am* 69, 8 (1979), 1183–1186.
- Barbara J Meier, Anne Morgan Spalter, and David B Karelitz. 2004. Interactive color palette tools. *IEEE Comp. Graph. App.* 24, 3 (2004), 64–72.
- Sára J. Molčan. 2016. I know what you're thinking... <https://www.instagram.com/p/BjOKKAKj7Mw/>. (2016). Instagram.
- Klaus Mueller, Joachim Giesen, Peter Zolliger, Lujin Wang, and Kevin T. McDonnell. 2008. Color Design for Illustrative Visualization. *IEEE Trans. Vis. Comp. Graph.* 14 (2008), 1739–1754.
- Chuong H. Nguyen, Tobias Ritschel, and Hans-Peter Seidel. 2015. Data-Driven Color Manifolds. *ACM Trans. Graph.* 34, 2 (March 2015), 20:1–20:9.
- Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2011. Color Compatibility from Large Datasets. *ACM Trans. Graph.* 30, 4 (July 2011), 63:1–63:12.
- Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Collaborative Filtering of Color Aesthetics. In *Proc. Wksp. Computational Aesthetics (CAe)*. 33–40.
- Yoshio Okumura. 2005. *Developing a spectral and colorimetric database of artist paint materials*. Master's thesis. Rochester Institute of Technology.
- Michael W Schwarz, William B Cowan, and John C Beatty. 1987. An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models. *ACM Trans. Graph.* 6, 2 (1987), 123–158.
- Alvy Ray Smith. 1978. Color gamut transform pairs. In *Proc. Conf. Computer graphics and interactive techniques (SIGGRAPH)*. 12–19.
- Alvy Ray Smith and Eric Ray Lyons. 1996. HWB—A More Intuitive Hue-Based Color Model. *J. Graph. Tools* 1, 1 (Jan. 1996), 3–17.
- KyounghHee Son, Seo Young Oh, Yongkwan Kim, Hayan Choi, Seok-Hyung Bae, and Ganguk Hwang. 2015. Color Sommelier: Interactive Color Recommendation System Based on Community-Generated Color Palettes. In *Adj. Proc. ACM Sym. User Interface Software & Technology (UIST Adjunct)*. 95–96.
- Peter Vandoren, Tom Van Laerhoven, Luc Claeßen, Johannes Taelman, Chris Raymaekers, and Frank Van Reeth. 2008. IntuPaint: Bridging the gap between physical and digital painting. *Proc. IEEE Int. Wksp. Horizontal Interactive Human Computer Systems (TABLETOP)* (2008), 65–72.
- Martijn Wijffelaars, Roel Vliegen, Jarke J. Van Wijk, and Erik-Jan Van Der Linden. 2008. Generating Color Palettes using Intuitive Parameters. *Computer Graphics Forum* 27, 3 (2008), 743–750.
- Brian Wyvill and Geoff Wyvill. 1989. Field functions for implicit surfaces. *The Visual Computer* 5, 1 (1989), 75–82.