Color Builder: A Direct Manipulation Interface for Versatile Color Theme Authoring

Maria Shugrina
University of Toronto

Wenjia Zhang
University of Toronto

Fanny Chevalier
University of Toronto

Sanja Fidler
University of Toronto

Karan Singh
University of Toronto

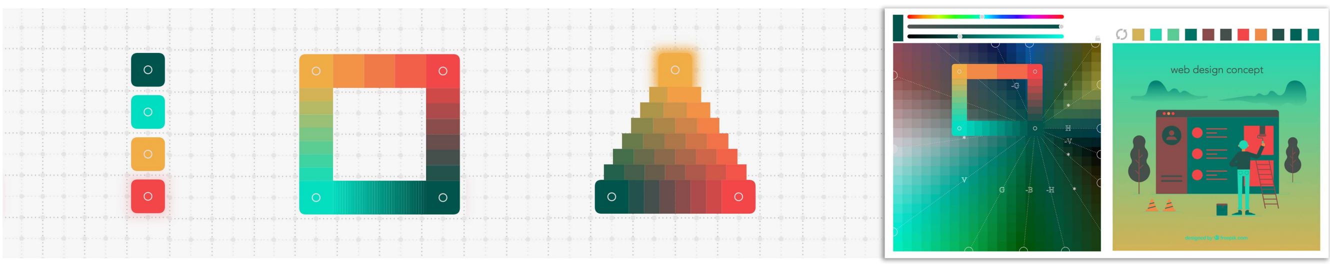


Figure 1: Designers can experiment with swatches, gradients and three-color blends in a unified Color Builder playground.

ABSTRACT

Color themes or palettes are popular for sharing color combinations across many visual domains. We present a novel interface for creating color themes through direct manipulation of color swatches. Users can create and rearrange swatches, and combine them into smooth and step-based gradients and three-color blends – all using a seamless touch or mouse input. Analysis of existing solutions reveals a fragmented color design workflow, where separate software is used for swatches, smooth and discrete gradients and for in-context color visualization. Our design unifies these tasks, while encouraging playful creative exploration. Adjusting a color using standard color pickers can break this interaction flow with mechanical slider manipulation. To keep interaction seamless, we additionally design an in situ color tweaking interface for freeform exploration of an entire color neighborhood. We evaluate our interface with a group of professional designers and students majoring in this field.

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300686

CCS CONCEPTS

• **Human-centered computing → Graphical user interfaces**; **User interface design**; *Web-based interaction*; Touch screens;

KEYWORDS

Color themes, color palettes, color pickers, gradients, direct manipulation interfaces, creativity support.

ACM Reference Format:
Maria Shugrina, Wenjia Zhang, Fanny Chevalier, Sanja Fidler, and Karan Singh. 2019. Color Builder: A Direct Manipulation Interface for Versatile Color Theme Authoring. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 12 pages.https://doi.org/10.1145/3290605.3300686

1 INTRODUCTION

Color has been shown to impact human cognition and behavior, as well as influence emotion [34]. Unsurprisingly, color plays a central role in many creative visual domains, including graphic design, visualization, user interface design, illustration, and beyond. Because of its importance and its subjective nature, the choice of color is not always obvious or immediate. To quote "Designer's Dictionary of Color" [1]:

> *Color is an incredibly powerful tool – but the task of choosing well can often leave designers stumped.*

Perhaps for this reason there are dozens of books on color inspiration [1, 33, 46, 78], as well as websites for creating, browsing and sharing palettes (also known as color themes) [3, 25]. In this work, we develop a novel interface supporting this popular task of color palette authoring and editing.

Most interfaces for color theme authoring are limited to creating small sets of independent swatches. This format becomes unwieldy as the number of swatches grows and ignores the common pattern of gradients and spatial blends, which are becoming popular in modern designs [76]. Gradient interfaces exist in separate software, forcing the designer to jump between browsing and editing swatch palettes, creating gradients from chosen swatch colors, and then using graphic design software to visualize these choices on their artwork – only to repeat the process when a color needs adjustment. This fragmented workflow may interrupt the psychological state of flow [54] and inhibit playful exploration, important components of the creative process [81].

Our work unifies authoring of swatch palettes, continuous gradients, step gradients and spatial blends in a single direct manipulation Color Builder interface. Color Builder works with mouse or touch input and is designed for play and freeform exploration, qualities important for creative interfaces [69]. Unlike most alternatives, our interface allows artists to rearrange palette layout and visualize colors relative to each other, reported important by artists and designers [44]. In addition, our tool supports in situ visualization of selected colors on an imported design, allowing exploration of color themes within a unified experience. Our design is rooted in analysis of existing software and workflows.

To facilitate seamless color exploration with direct manipulation we additionally propose an in-context color picker optimized for fine color adjustment, a common task during color theme authoring. Unlike the standard color pickers, which confine the user to pre-defined color space axes, our design displays a comprehensive map of the color's local neighborhood and allows single-swipe tweaking of the color within this space. Our color picker is integrated into Color Builder, but can also function independently, complementing standard color pickers available today.

Specifically, the contributions of this work are as follows:
- Analysis of existing color theme workflows/software (§3)
- Novel versatile direct manipulation color theme authoring interface that unites currently disjoint color workflows (§4,5) and supports interactive visualization (§6)
- Novel contextual color picker design (§7)

We evaluate the Color Builder design and the contextual color picker with a group of design professionals in §8.

2 RELATED WORK

Discrete Color Palette Interfaces: Discrete color palettes (Fig.2A), comprised of independent color swatches, are a ubiquitous representation of color combinations in design

Figure 2: Common types of color combinations: A - discrete swatch palette[1] from [25], B - continuous gradient from [32], C - discrete, step-based gradient with colors from [16].

applications. Popular software, such as Adobe Illustrator and Sketch [13], allows users to create swatch palettes by manually adding individual colors. These swatch palettes or libraries can be shared in a variety of formats (e.g. .aco, .ase, .swatches). In addition, a number of online apps allow users to construct, share and rate small (5-10 colors) swatch palettes [3, 11, 21, 23–26, 37, 59, 64, 68, 84]. Its simplicity has made swatch palette popular not only in consumer software, but also academic research, including applications in automatic palette extraction from images [15, 47, 56], image recoloring [15, 18, 79] and colorization [48], crowd-sourcing color aesthetics [56, 57], and machine learning [23]. Today, many commercial applications also allow automatic palette construction from an image [3, 11, 14, 25, 64]. Swatch palettes are limited to use cases where the number of relevant colors is small, excluding common cases such as color gradients.

Gradient Interfaces: Gradients or transitions between two or more colors (Fig.2B,C) are a common tool in user interface (UI) and graphic design, as well as illustration, where shading naturally introduces gradients. A number of recent design blogs have observed that gradients are becoming more popular [53, 67, 76], following the trend set by Instagram's 2016 redesign [74]. Gradient appearance is hard to predict from endpoint colors alone, and merely saving endpoint colors in a swatch palette loses connection to the actual colors in the artwork. While gradient construction and visualization is not supported by existing swatch palette interfaces, specialized websites allow working specifically with gradients [7, 32, 35, 43, 58, 66, 75, 80][2]. Smooth gradients (Fig.2B), of course, can also be created in many established graphic design applications, such as Adobe Illustrator, but they cannot be saved and accessed as a part of a reusable palette. In addition, complicated series of steps may be required to automatically construct discrete step-based gradients (Fig.2C), as illustrated here [31]. Yet, discrete gradients are not an uncommon use case, and some specialized tools allow directly constructing them [16, 22, 62] or generating suggested step gradients from a single seed color [36, 66].

[0] Available at http://www.colourlovers.com/palette/92095/Giant_Goldfish.

[2] Many of these can be browsed from [8].

Spatial Blend Interfaces: A natural extension of a linear gradient is a general spatial color blend, a concept with no established representation. Some interfaces model continuous palettes by directly simulating paint [2, 10, 12, 77], or by approximating it with simplified color blobs [70]. Some approaches automatically fit continuous blends, which cannot be directly edited [55]. While compelling for their respective use cases, these interfaces focus on the continuous case and do not naturally support discrete colors. Our goal is to provide a structured representation spanning discrete and continuous palettes, while allowing interactive editing.

Multi-Function Creative Color Interfaces: Limited prior research takes a holistic look at designing interfaces for diverse color needs. Meier et al. propose a number of separate interfaces for browsing swatch palettes, mixing, spatial arrangement, compositing and visualization [51]. Jalal et al. take a close look at a diverse set of digital and physical color workflows employed by artists and propose a design space for color tools, evaluating it on a number of trial interfaces [44]. We use insights from these broad explorations as we focus on optimizing our interface design for the specific task of constructing reusable and versatile color themes. Other research has optimized color interfaces for specific *creative domains*, such as digital painting [2, 10, 12, 70, 77]. In contrast, our goal is to make our interface broadly applicable across all visual domains where color themes are relevant.

Suggestive Color Palette Interfaces: A separate class of interfaces provide smart assistance for palette authoring. Many interfaces construct palettes based on color theory constraints [3, 51, 59] or its extensions [40]; others offer more flexible constraint formulations[52]. Other methods allow semi-automatic palette construction based on observations about color perception [82] or color naming [39]. Data-driven methods complete, generate or interpolate discrete palettes by learning from palette collections [23, 56, 63, 73]. Other software generates discrete color themes from seed colors based on proprietary principles [11, 21, 64, 66]. We believe such capabilities are complementary to our work and could be incorporated into the interface we propose.

Color Space Representations: There are many alternatives to the red-green-blue (RGB) digital color space. Hue-value-saturation (HSV) [71] and hue-white-black (HWB) [72] spaces are designed for intuitive control. Spaces such as CIELAB (Lab) and CIEDE 2000 [28] are more perceptually uniform, while others are tailored to human color perception [29, 50]. Chosen color space has a significant effect on linear color interpolation, and also many non-linear blending models are available, including modeling paint layers [38], learning blending from data [49], or deriving aesthetically pleasing interpolation curves [27, 82]. Our gradients use RGB interpolation, most common in graphic design software, and our color picker relies on RGB, HSV and Lab spaces.

Color Picker Interfaces: Color pickers allow selecting or adjusting a color. Ubiquitous color pickers available today have not changed for decades [44] and include the HSV color wheel, RGB sliders, and 2D color space projections accompanied by a slider. One inventive design [37] maps HSV space to mouse controls (X,Y,scroll), but provides no visual guidance. Alternative interfaces allow picking colors from user-created blends [27, 60, 70]. Some interfaces provide color suggestions by mining compatible colors from discrete palette collections [56, 73] or using randomization [11]. Other interfaces show a selection of color alternatives similar to a base color by sampling along hue and value directions in HSV, or along analogous and complementary hue directions, stemming from color theory [41, 84], or using proprietary algorithms [26, 61, 64]. These interfaces either provide only a very limited suggestion pool [41, 61, 64, 84] or present a redundant and unorganized grid of colors [26] (See Fig.7b). Similarly, we develop a contextual color picker for *adjusting* a starting color within its neighborhood, but we examine all available color adjustment directions and provide comprehensive and easy-to-navigate choices.

3 MOTIVATION AND DESIGN CHOICES

The space of color interfaces is vast. We focus on the task of constructing, editing and exploring reusable color themes for a range of design disciplines. To understand current practices, we distributed an online survey (see Appendix) to a wide list of designers, and collected responses from 7 graphic/UI design and illustration professionals (4 practicing full-time, 3 part-time).[3] In addition, we analyzed a number of online video tutorials and blogs about color and palettes (cited below). Our analysis revealed that existing interfaces (Table 1) address specific use cases and do not provide a full solution, and prompted us to formulate the following design goals.

Goal 1: Unify Swatches and Gradients

Swatch palettes and gradients are important components of color design. Swatch palettes help organize design colors[17, 30] and thousands of them are shared online [3, 25], speaking to their popularity. These palettes do not naturally support gradient colors. Many online design tutorials explain how to work with gradients [20, 31, 45], and several blogs observe that color gradients are back in vogue [53, 67, 76]. To get anecdotal evidence about this claim we examined the Behance Curated Graphic Design Gallery [4], selecting the first 100 non-photograph designs with more than 2 colors. Out of these 100, 25 had at least one continuous gradient and 10 contained colors derived from a step gradient. 6/7 designers we surveyed indicated that they use gradients and said their workflow has room for improvement.

[3] This group is disjoint from our user study participants.

	DS	CG	SG	SB	LR	VD	DM
1. Design Suites							
Adobe Photoshop	✓	✓	✗	P	P	✓	✗
Adobe Illustrator	✓	✓	✗	✓	P	✓	✗
Sketch [13]	✓	✓	✗	✗	P	✓	✗
Paper 53 [60]	✓	✓	✗	✗	P	P	P
2. Palette Tools							
Paletton [59]	✓	✗	P	✗	✗	✓	✗
ColorRotate [42]	✓	✗	✗	✓	✗	P	✓
colormind.io [23]	✓	✗	✗	✗	✗	✓	✗
Adobe Color [3]	✓	✗	✗	✗	✗	✗	P
ColourLovers [25]	✓	✗	✗	✗	P	✗	✗
Coolors [11]	✓	✗	✗	✗	✗	✗	✗
Colrd Palette [26]	✓	✗	✗	✗	✗	✗	✗
Color Munki [84]	✓	✗	P	✗	✗	✗	✗
Color Blender [21]	✓	✗	✗	✗	✗	✗	✗
ColorSpire [24]	✓	✗	✗	✗	✗	P	✗
Colordot [37]	✓	✗	✗	✗	✗	✗	P
ColorExplorer [64]	✓	✗	✗	✗	✗	✗	✗
3. Gradient Tools							
Grabient [75]	✗	✓	✗	✗	✗	✗	P
GradPad [58]	✗	✓	✗	✗	✗	✗	P
CSS Gradient [7]	✗	✓	✗	✗	✗	✗	✗
Color Space [66]	P	✓	P	✗	✗	✗	✗
Material Design [36]	P	✗	P	✗	✗	P	✗
0to255 [16]	✗	✗	P	✗	✗	✗	✗
Gradient Maker [62]	✗	✗	✓	✗	✗	✗	✗
4. Research Tools							
Playful Palette [70]	P	✓	✗	✓	✓	P	✓
Mixer in [51]	✗	✓	✗	✗	✓	✗	✓
Composition in [51]	✓	✗	✗	✗	✗	✓	✓
Explorer in [44]	✓	✗	✗	✗	✓	P	✗
virtual paint [2, 10]	✗	✓	✗	✓	P	✗	P
Color Builder (ours)	✓	✓	✓	✓	P	✓	✓

Table 1: Palette Interfaces. DS: discrete swatches, CG: continuous gradients, SG: step gradients, SB: spatial blends, LR: layout rearrangement, VD: visualization on a design, DM: direct manipulation, ✓: full support, P: partial support, ✗: no support.

As shown in Table 1, the sets of tools supporting discrete swatch palettes (DS) and continuous (CG) or step gradients (SG) are nearly disjoint. Full-featured Design Suites do support smooth gradients, but not as a part of a reusable palette, requiring manual steps to apply swatch colors to the gradient. One designer we surveyed wrote: "I find Illustrator's color palette system frustrating sometimes. Particularly going between the color palette and gradient tools." Generating step gradients, especially common in UI design for effects like hover [45], requires either complex manual procedures in tools like Illustrator [31] or a specialized step generator [16, 22, 62]. If the seed colors change, the designer must recreate the intermediate colors in a separate software/workflow and manually re-import updated swatches into her palette. Our interviewees found this frustrating: "if my initial palette changes, I have to go back in and change [all swatches] individually," "I'd like to see an easier way to move between full color palettes, as opposed to adjusting colors one by one".

Lack of unified support for swatch palettes and gradients induces a fragmented workflow. Consider the progression in Fig.3A-F. The designer may choose an initial discrete palette (A), then use a separate step generator only to discover that the gradient is too faint (B). Necessary adjustments will require re-generation of gradient colors (C). If the designer then experiments with a smooth gradient using the palette colors, she may discover that the rose reflection (C) clashes with the white-red gradient (D). Finally, a more complex tri-color blend may render the whole palette unusable due to the blending behavior between dark teal and red (E), causing the designer to rethink the entire set of colors (F). As one designer states: "There isn't really a tool that does everything well." Our goal is to create a unified representation and authoring tool for swatch palettes and gradients, inviting exploration and sharing of more complex color choices.

Goal 2: Support Visualization and Layout

Color perception is inextricably linked to its context [6] and in-context visualization is important for color design tools [9, 44, 51]. The simplest visualization is rearranging the palette layout itself (Table 1, SR). Several palette tools allow swatch re-ordering or stretching (P), and some allow full spatial rearrangement and swatch scaling (✓). This form of visualization provides an important design-agnostic snapshot of the palette. However, colors may appear very different on a more realistic design. Most palette tools offer no visualization on a design (Table 1, VD), and some show colors mapped to regions of several fixed designs [23, 51, 59]. More customized visualization requires importing the palette into a full-featured Design Suite (Table 1.1) and applying colors by hand. Several designers we surveyed suggested that this can be disruptive: "Sometimes a colour viewed individually looks soft but when placed in the mockup, looks very salient (or vice versa). This means going back to tweak it then repeating the process again." Our aim is to support both layout rearrangement (LR) and visualization on a *custom* design (VD), inviting uninterrupted color theme exploration.

Goal 3: Encourage Play and Exploration

Ability to play and explore has been shown important for interfaces supporting creativity [19, 65, 69] and for inducing absorption in the task or flow [5]. Our goal is to design an interface that is interactive, fun and immersive. Specifically, we strive for simple direct manipulation interaction devoid of specialized and over-constrained widgets, menus, sliders and similar. Our goal is an interface that feels like a game and

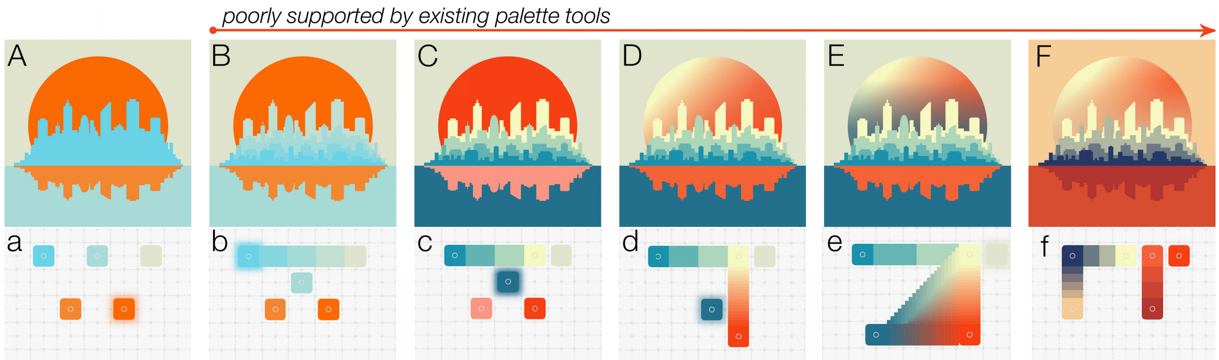
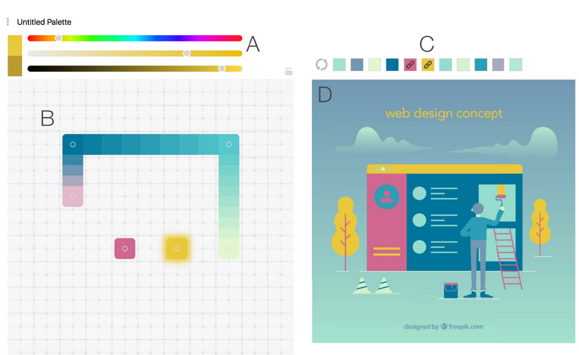
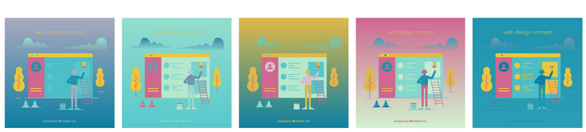


Figure 3: Existing palette tools fail to support color design use cases B-F that go beyond discrete swatch palettes (A). Color Builder supports a range of palettes from swatches (a), to discrete (b,c) and continuous (d,f) gradients and tri-color blends (e), while providing custom visualization on an imported vector artwork (A-F). Skyline sourced from an image by Freepik.

encourages freeform exploration and reflection-in-action, another important concept in design [83]. Some tools follow this design principle of exploration through direct manipulation (Table 1, DM). In particular, Playful Palette [70] and the mixer in [51] allow playing with color circles and continuous gradients. However, these tools do not support discrete swatches or step gradients, and have limited visualization. Other tools support playful exploration for specific tasks of authoring swatch palettes [11] or gradients [58]. We apply the same principles of playfulness and direct manipulation to design an interface supporting all features in Table 1.



(a) Screenshot of our touch-optimized Web interface.



(b) Random visualizations with some regions locked.

Figure 4: The Color Builder interface (a) allows interactive palette construction and visualization on a vector design (image by Freepik). Users can cycle through random assignments of colors and lock individual color regions (b).

Goal 4: Facilitate In-Context Color Adjustment

Undoubtedly, selecting colors for a color theme is critical, but adjusting initial selections is at least as important. As one of the surveyed designers puts it, it is common to "pick everything, and then realize none of it looks right", while the other reports: "I love being able to go OCD over my colors and fiddle with the saturation or hue values." Canonical color pickers (§2, Fig.7a) provide fast access to the entire color space, but may not always excel at finer adjustments. One of the users we interviewed observed that a "very small movement" on the HSV color wheel results in a completely different color, and another said that "granularity is an issue when the color picker interface is small." A third participant responded that for fine adjustment "the global color edit system is not very intuitive or helpful." While we resort to classical HSV color sliders for initial color selection, our goal is to facilitate in-context color adjustment that follows the philosophy of playfulness and direct manipulation (above).

4 SYSTEM OVERVIEW

The Color Builder system is designed for creating complex color palettes of swatches, gradients and tri-color blends (Goal 1) using direct manipulation (Goal 3), while providing quick visualization (Goal 2) and a way to quickly tweak chosen colors (Goal 4). Our interface includes the main Color Builder active grid (Fig.4a.B), where swatches can be added, combined and rearranged (See §5). On the right, we provide an optional preview pane (Fig.4a.D) where a fixed mockup or user's own SVG vector graphic can be imported for quick visualizations (Fig.4b); see §6. The color of a selected swatch (glowing) can be changed with standard HSV sliders (Fig.4a.A), or using an in-context Color Tweaker (§7). Our prototype is a touch-optimized HTML5 Web implementation, with all interactions defined for both mouse and a touch screen, where direct manipulation may have a more natural feel.

5 DIRECT MANIPULATION INTERACTIONS

Color Builder supports creating and visualizing swatches, step gradients, visually smooth gradients and tri-color blends using direct manipulation swatch interactions on an active palette grid (Fig.4a.B). Consider the progression in (Fig.3). The designer may first create the discrete palette (Fig.3a) by tapping/clicking on empty areas to add swatches (Fig.5a). The selected swatch, indicated by a glow, (Fig.5b) can be recolored with HSV sliders or Color Tweaker (§7), dragged around the screen while snapping to grid (Fig.5d) or dragged off screen to be deleted (Fig.5e). Swatches explicitly created by the user have a circle handle and are referred to as *controllable swatches*. The designer may then import an SVG graphic and apply the palette to it (Fig.3A); see §6.

Next, the designer may decide to create a step gradient in Fig.3b. She does this by dragging one *controllable swatch* s_0 next to[4] another *controllable swatch* s_1 (Fig.5f). When s_0 touches s_1, it "sticks" and dragging it in the opposite direction automatically creates a gradient with the number of steps proportional to how far it is dragged (Fig.5g). Once the gradient is created, the designer increases/decreases the number of steps by touching/clicking intermediate gradient colors and dragging up/down (Fig.5h). After applying the 5 gradient colors in the step gradient to the visualized artwork (Fig.3B), the designer realizes that she needs more contrast.

She taps *controllable swatches* at the gradient endpoints to select (Fig.5i) and recolors them as any other *controllable swatches*, automatically affecting the intermediate colors and the city in the visualization (Fig.3C,c). She may also recolor the orange swatches to coordinate with the new gradient. She can reorganize her palette by dragging the endpoints of the gradient to lengthen or shorten it or to move it up/down; all gradients are constrained to stay horizontal or vertical (Fig.5j). Finally, if she later decides to delete the gradient, she can drag one of the endpoints close to the other, activating deletion markers (Fig.5k). If she releases touch/mouse in this state the gradient will be deleted, but its endpoints will remain. The designer may attach as many as 4 gradients to any *controllable swatch* and create visually continuous[5] gradients (Fig.3d), which may prompt her to select a coordinating color for the city reflection from the smooth blend (Fig.3D).

Finally, the designer may experiment with tri-color blends and discover that her color choices do not look good blended (Fig.3E,e). Blends work much the same as gradients, using barycentric coordinates of intermediate swatches for linear RGB interpolation of 3 colors. If a *controllable swatch* s_2 is dragged near the intermediate swatches in an s_0-s_1 gradient, it "sticks" to the gradient and dragging in the opposite direction lifts a pyramid of intermediate colors with the same step number as in the base gradient (Fig.5l,m). As with gradients, endpoints of a tri-color blend can be recolored, and the number of steps can be changed by touching inside and dragging up/down (Fig.5n). Dragging one of s_0, s_1, s_2 near its neighbor will show deletion indicators and releasing in this state will cause the blend to be deleted (Fig.5o). When dragging the endpoints, the base gradient is constrained to stay horizontal/vertical and s_2 adjusts to avoid obtuse angles. Blends can have horizontal or vertical orientation, and *controllable swatches* always snap to the grid.

[4]Our interface implements collision detection to prevent swatch overlap.
[5]High subdivision visually approaches a smooth gradient, and our implementation does not explicitly transition from a step to continuous gradient, but could be amended to automatically switch to smooth rendering.

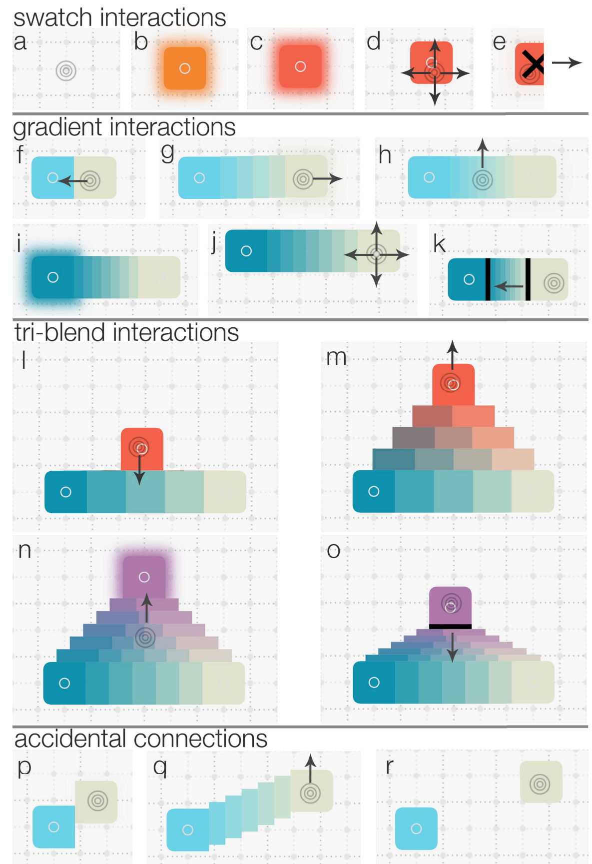


Figure 5: Swatch interactions for creating (a,f-g,l-m), moving (d,j,o) and deleting (e,k,o) swatches, gradients and blends. Gestures can change gradient/blend steps (h,n). Auto-snap prevents accidental connections(p-r).

Swatches stick on contact, and to prevent accidental gradients/blends we equip just-formed gradients/blends with "auto-snap", which automatically destroys them if the dragged swatch travels beyond their valid geometric configuration. E.g. if s_0 in Fig.5p touches s_1, a gradient is created, but as s_0 continues to travel beyond horizontal gradient orientation (Fig.5q), the temporary gradient snaps automatically (Fig.5r).

6 VISUALIZATION

In addition to palette layout rearrangement, Color Builder allows importing any SVG vector design for quick visualization. The system automatically groups SVG colors and displays them in the Linking Pane (Fig.4a.C, Fig.6A). The user can then cycle through random assignments of palette colors to the image by pressing Fig.6A.2, or manually link certain SVG color groups to specific palette colors using the Linking Pane (Fig.6A.1). This flexible visualization spans the range between full control and quick automatic preview.

Consider Fig.3E, where the colors do not work well in the tri-color blend. Suppose that the designer has mapped all the color groups in the Linking Pane to the colors in her test image (see link indicators in Fig.6A). She may try out different color combinations by recoloring a few *controllable swatches* in her palette, thus instantaneously recoloring the image (Fig.6B). At this point, she may feel that her manual selections are too bold. To get ideas for using more subtle shades from the blends in the palette, she can unlock some groups in the Linking Pane and press the cycle button repeatedly to get new ideas for the other colorizations, while keeping the city colors constant (Fig.6C). She can fully unlock the Linking Pane, and cycle through totally random assignments in Fig.6D. This effortless color cycling can provide a quick feel for the interplay of colors in a starter palette and help serendipitously discover successful blends.

7 IN-CONTEXT COLOR TWEAKER

It can be hard to find the fork in question on a folded trail map. In contrast, GPS applications automatically tailor the view to the current location. We develop an analogous experience for exploring the 3D color landscape: where standard color pickers show the global view, our color tweaker presents a contextual neighborhood map (Fig.7). A tap on a selected *controllable swatch* activates the neighborhood color map centered at that color and allows direct manipulation color adjustment within the Color Builder active grid. The designer can freely explore the map, recoloring the swatch, and tap the swatch when done. Pinch in/out gestures or mouse wheel zoom in/out on the neighborhood. This design allows adjusting colors without navigating to a separate widget.

Laying out the Map

3D color spaces (RGB, HSV, Lab) afford no obvious way to lay out a local map on a 2D screen. Standard color pickers project the view on 2 dimensions and add a slider for the third (Fig.7a). This can make it difficult to explore diagonal directions, and the hunt for the right adjustment may require multiple slider moves. Neighboring color interfaces (Fig.7b) show colors sampled near a base color, but present either a small number of discrete choices or an unorganized

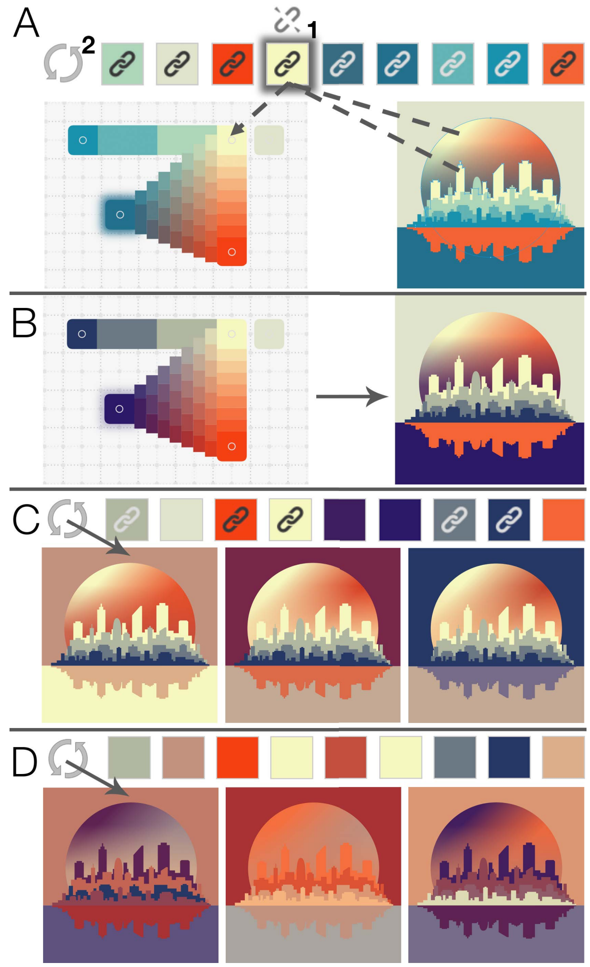
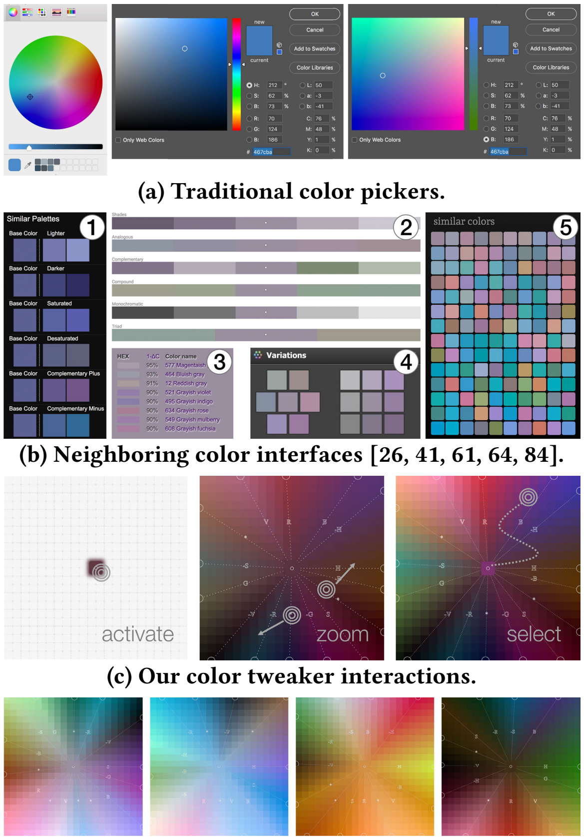


Figure 6: Visualization pane allows recoloring any input SVG using the palette colors with manual assignment (A,B), partially or fully random color assignments (C,D).

map. Expanding on these approaches, we select 3D "tweak" directions that comprehensively sample the space of possibilities, and lay them out radially around the source swatch in 2D, interpolating in between. The result is a single smooth map showing a rich variety of easily accessible local colors (Fig.7c). To match the grid layout we discretize the color map, and annotate it with tweak directions for guided navigation.

To find viable tweak directions $\mathbf{D} = \{\boldsymbol{d}_0 \ldots \boldsymbol{d}_k\}$, we first generate 12 neighbors around the source color $\boldsymbol{c}$ by incrementing and decrementing dimensions in HSV and RGB. This yields 12 vectors, which we represent in the more perceptually uniform Lab space. These standard exploration directions are familiar to users and provide a good starting point, but may not sample the neighborhood equally well for all colors (Fig.8): e.g. -G and -H directions may coincide. To find augmenting directions, we uniformly sample vectors



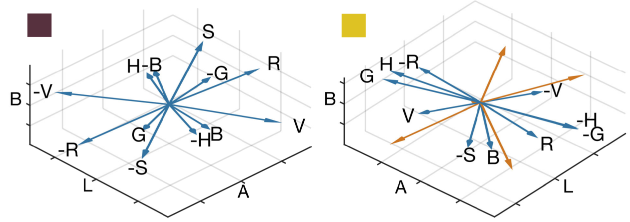
(a) Traditional color pickers.

(b) Neighboring color interfaces [26, 41, 61, 64, 84].

(c) Our color tweaker interactions.

(d) Neighborhood maps for various colors.

Figure 7: Unlike traditional pickers (a), contextual color tweaker shows a full view of a color's local neighborhood.



(a) seed $c = rgb(89, 50, 64)$ **(b) seed $c = rgb(223, 195, 35)$**

Figure 8: Standard directions in RGB, HSV (blue) can have a good distribution over the possibilities (a), or require augmenting directions (orange) (b). Plotted in Lab space.

around $\boldsymbol{c}$ and pick a subset by furthest point sampling.[6] We add samples to $\mathbf{D}$ until it reaches size $k = 15$.[7] The augmenting directions improve the sampling of the neighborhood (Fig.8b), resulting in more comprehensive visualizations.

Once $\mathbf{D}$ is complete, we convert each vector $\boldsymbol{d}_i$ in Lab to a neighboring color $\boldsymbol{a}_i$ by adding $z_i \boldsymbol{d}_i$ to the source color $\boldsymbol{c}$, where z_i is the zoom value. While the color map has a global zoom level z, we use binary search to find the largest $z_i \leq z$, such that $\boldsymbol{a}_i$ falls within the valid color gamut. As some directions may truncate early, we discard those $\boldsymbol{a}_i$ that fall within τ of each other or $\boldsymbol{c}$ in the Lab space (we use $\tau = 8.0$). We find a smooth ordering of the neighbors $\boldsymbol{a}_i$ using a greedy approximation of the Traveling Salesman Problem (TSP) over $L2$ Lab distances. We lay out the ordered colors radially around the source swatch, snapping each to the grid edge. The angles are proportional to the smoothed Lab distances between $\boldsymbol{a}_i$'s. Linear RGB interpolation within each wedge produces a smooth blend of alternatives, and each direction is annotated with its meaning (Fig.7d). Interactively changing the zoom z affects the range of displayed possibilities. Our algorithm runs in real time, and users can even keep the neighborhood map open and see it change while adjusting the starting color with HSV sliders (Fig.4).

[6] We use arc distance on a unit sphere for normalized vectors in $\mathbf{D}$.

[7] The number of additions varies, as the starting set may have duplicates.

8 USER EVALUATION

We conducted an expert evaluation of the Color Builder interface with 10 professionals (4 male, 6 female) practicing in the fields of graphic design (6 people), illustration (3) and UI/UX design (1). Six participants had more than 5 years of professional experience in these fields, while others had 1-4 years experience and majored in one of these fields. All rated familiarity with Adobe Photoshop or Illustrator as at least 4 on a 5-point scale and reported their ages as 18-24 (3), 25-34 (4), and 35-44 (3). To recruit, we distributed fliers on an art college campus and in art stores. The study lasted 1 hour and compensated participants with $25 art supplies gift cards.

User Study Design

This study aims to evaluate three aspects of Color Builder:

A1 overall effectiveness at color theme authoring

A2 contextual color tweaker

A3 linked and random color visualization on a design

To this effect, we designed two tasks, each administered in two conditions: **condA** using Color Builder interface, and **condB** using commercially available alternatives.

Task 1 focused on A1. Each participant was asked to come up with a hypothetical design project (e.g. a party poster) and design a starting color theme for it using Color Builder(condA) and Paletton[59](condB). The condition order was reversed for half of the users, and task time was capped at 7 minutes per condition. For the second condition, we asked participants to think fresh, not replicate the first color theme. Each condition was prefaced with a training session for that interface, followed by a quick quiz to ensure users recalled the functionality. Thus, users with BA ordering were introduced to Color Builder after performing Task 1 in condB. The training session for Color Builder took 5-7 minutes and less for Paletton, which has fewer features. We administered the Creativity Support questionnaire (below) throughout the task and qualitative questions at the end.

For **Task2**, we focused on A2 and A3 and asked users to edit a color theme linked to a design. In condA, participants used a combination of the contextual color tweaker and the HSV sliders in Color Builder to adjust the color of a swatch and a gradient linked to a visualization. In condB, users worked with Adobe Photoshop color picker and Illustrator swatch panel linked to the same design. Up to 5 minutes were allocated to each condition. We used the ordering from Task 1, and concluded with qualitative questions.

We chose Paletton for our comparison in Task 1, because it is a dedicated palette tool, like our interface, and supports several dimensions in Table 1. Comparing against Design Suites would not be fair or meaningful, and exact counterparts among palette tools do not exist. For Task 2, Photoshop color picker is the industry standard and a natural choice, and Illustrator swatch panel is one of few tools able to edit color themes linked to an image.

For both tasks, condA was on an iPad pro and condB on a laptop computer with a mouse. The Color Builder interface was designed with playful touch interaction in mind, and we evaluate it in this setting. We chose the most comparable applications for condB, but they were not optimized for/available on a tablet. While the difference in input modalities can introduce bias, we believe that this bias is meaningful: if users prefer touch interaction, the fact that our interface is optimized for touch should play to its advantage, especially given the lack of comparable alternatives.

Quantitative Results

We evaluate Color Builder's support for the creative task of color theme authoring using the Creativity Support Index (CSI) [19], which uses a research-tested questionnaire to rate creativity support of an interface using 5 factors: Collaboration, Enjoyment, Exploration, Expressiveness, Immersion and Results Worth Effort. The CSI assigns an overall score of 0 (worst) to 100 (best), and is most meaningful when comparing two interfaces on a specific task. We administer CSI questionnaire after each condition in **Task 1**, and compare results for Color Builder and Paletton [59].

Color Builder performed considerably better, achieving the mean CSI score of 71.8 (std=14.0), compared to the score of 44.8 (std=21.7) for Paletton. To estimate statistical significance of this result, we used a one-tail paired samples t-test with the null hypothesis that the true CSI means are the same. The t-test rejected the null hypothesis with a low p-value of 0.0067. It is also meaningful to look at the performance of the interfaces for specific CSI factors (Table 2).[8] Color Builder outperformed Paletton most consistently in Expressiveness. Like most palette tools, Paletton is restricted to a small number of colors, while Color Builder allows much more expressive palettes (Fig.9). Our users created 4-22 controllable swatches (mean 7.2), and all but one made at least one gradient/blend. Six users added 2-6 gradients, and four experimented with three color blends. Color Builder also scored consistently better in Exploration, likely due to its freeform nature and the exploratory color tweaker. Better performance on Enjoyment is consistent with users' feedback (below). Improvement in Immersion and Results Worth Effort was less statistically significant, which could be due to the short task duration and the learning curve.The concept of flow is incorporated into several CSI factors, notably Immersion and Expressiveness ([19], Sec 4), suggesting that Color Builder may be more successful at eliciting this state.

CSI Factor	Paletton Score	ColorB. Score	T-test result
Expressiveness	32.4	50.2	$p = 0.0129$
Enjoyment	20.3	31.6	$p = 0.0170$
Exploration	35.8	55.5	$p = 0.0173$
Immersion	15.8	29.2	$p = 0.0378$
Results Worth Effort	29.2	43.0	$p = 0.0983$
Overall CSI	44.8	71.8	$p = 0.0067$

Table 2: Break down of the CSI scores by factor with corresponding one-tail paired-samples t-test.

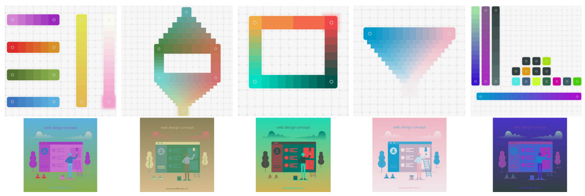
Qualitative Feedback

8/10 users said they create color themes in their work. When asked when they would prefer Paletton and when Color Builder for this task and why, 5 users unequivocally preferred Color Builder, 2 would use both, 2 said that neither fulfills the requirements of their workflow, and 1 preferred Paletton.

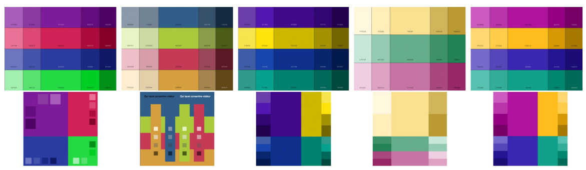
Exploration and Play: Users who preferred Color Builder stressed its support for exploration and play, and the speed of experimentation: "It's easier, it's faster. It gives me more freedom to play with colors" (user I), "It's more fun. If you are being creative, you want things to be more playful" (user C), "Such a fun interface, almost like a game, playing with these little blocks" (H), "[This] is the only color theme tool I have used yet that actually worked. Have not enjoyed ANY of the other ones I have tried" (G). Even user F, who preferred Paletton, said that Color Builder is "More fun to play with", but found the interface to be too unconstrained.

Color Builder Interactions: All users easily picked up touch interactions during the training session, and several commented that they found them intuitive, e.g.: "The simple motions used to blend and create swatches made this interface easy to use" (A). 8/10 users said that gradients are important in their work, and found the ability to create them when authoring a color theme very useful, as it is hard to tell if solid swatches would work well in a blend. E.g. H found step gradients useful for 3-step shadows, and E explained

[8] We omit Collaboration, as most users marked it as not applicable.



(a) Palettes created in Task1, condA.



(b) Palettes created in Task1, condB (same users).

Figure 9: Sample palettes, visualizations from the user study.

that for illustration 3-color blends naturally span the shadows, midtones and highlights of a shaded object. Due to variation in touch styles, the gesture for adjusting gradient subdivisions did not register consistently for 2 users, and an additional way to control this setting would improve usability. We also found that the HSV sliders above the palette pane should be larger for comfortable finger interaction.

Constraints: Design constraints was the main concerns brought up by our participants. Many stressed the importance of numeric color entry and pinning client's brand colors, remedied by small additions to our interface. Two users who preferred neither interface work within narrow fabrication/printing constraints, not handled by Color Builder, Paletton or most tools. Several users found color theory constraints in Paletton useful: "[this] removes the frustration of finding complementing colors" (F). User F compared these constraints to "training wheels" for color themes, but 6 users also cited them as too limiting. There is a fine balance between freeform play and design constraints. In this work, we focus on the latter, and handling of general constraints in Color Builder presents a challenging future direction.

Visualization: Most users found visualization useful and said it plays well with the incremental nature of design. 7/10 users used random color mapping during Task 1, cycling on average 11.4 times, and 5 said that ability to visualize on a realistic design is critical for designing a theme: "I like that this applies it (palette) before I put in a bunch of work designing something" (D). Several users found the "learning curve" for the Linking Pane too steep for the allotted time, and focused on using random color assignment. Visualization is not the main feature of our interface and we defer its deeper exploration to future work.

Color Tweaker: Our color tweaker is intended as a complementary interface for adjusting a color in its local context. Most users found this useful: "[It] allowed me to change color without loosing the 'feel' of the original" (G), "I was able to play with it and learn from playing with it" (C). Photoshop picker was preferred when the required adjustment was clear (e.g. decrease saturation), or when a wildly different color was needed. Most designers said that the color tweaker is preferable when the choice of a color is unclear, e.g. when looking for a color that functions well within a design or a blend: "When I am adjusting multiple colors, I need to think what my options are – choice is less obvious" (D). Immediate visualization coupled with the color tweaker worked well in this case. Users engaged with the tweaker without being explicitly asked: on average 6.9% of Task 1 (condA) duration was spent with the color map open (third of this time actively selecting colors). This is comparable to the 5.9% time spent interacting with the HSV picker, best for different use cases. Our participants had a wealth of experience using standard color pickers, yet 8/10 said they would incorporate the color tweaker into their workflow if it were available.

3 users found the overlay of the current palette on the color map particularly useful to see adjustments in context. At the same time, presenting both color theme and color map in the same space caused problems when the palette occupied much of the grid, or when the seed color was close to the palette edge, causing color map to be squashed in that area. More experimentation is needed to find a solution that both juxtaposes the map with the current palette and allows unobscured view of the map. While users appreciated the variety of randomized colormaps, some found it disconcerting that the map was lost once closed. A way to remedy this would be one user's suggestion to save color maps for future use. Other missing capabilities included exploring further in a particular direction and zooming in on an area of interest. 5 users mentioned that discrete cells of the map give them more confidence: Photoshop picker provides no way to save alternatives, while our static discrete color map allows one to keep a mental note of viable options during exploration.

9 CONCLUSION & FUTURE WORK

We presented Color Builder, a unified interface for constructing color themes of swatches, gradients and three color blends in a flexible and playful way, while providing in-context visualization, and a novel in-context color tweaker. A group of professional users found many aspects of our interface not only useful, but also fun, playful and exploratory.

In future work, we would like to address alternative color interpolation, including HSV, Lab and user-controlled blending. In addition, we would like to thoroughly examine user needs for color theme visualization, for which we only provide one solution in the current prototype. Further, we would like to address user feedback and explore ways to incorporate color theory and other constraints (e.g. fabrication) into the freeform and playful Color Builder interface.

REFERENCES

[1] Sean Adams. 2017. *The Designer's Dictionary of Color.* Abrams Books.
[2] Adobe. 2011. Adobe Color Lava for Photoshop. https://www.youtube.com/watch?v=iF4e5bHPD24.
[3] Adobe. 2018. Adobe Color CC. https://color.adobe.com.
[4] Adobe. 2018. Behance Curated Graphic Design Gallery. https://www.behance.net/galleries/2/Graphic-Design?content=projects&queues=87. Accessed: 2018-08-21.
[5] Ritu Agarwal and Elena Karahanna. 2000. Time flies when you're having fun: Cognitive absorption and beliefs about information technology usage. *MIS quarterly* (2000), 665–694.
[6] Josef Albers. 2013. *Interaction of Color* (50th anniversary edition ed.). Yale University Press.
[7] Moe Amaya. 2018. CSS Gradient. https://cssgradient.io/.
[8] Moe Amaya. 2018. Gradient Backgrounds. https://cssgradient.io/gradient-backgrounds/.
[9] Penny F Bauersfeld and Jodi L Slater. 1991. User-oriented color interface design: direct manipulation of color in context. In *CHI.* 417–418.
[10] Bill Baxter, Vincent Scheib, Ming C Lin, and Dinesh Manocha. 2001. DAB: Interactive haptic painting with 3D virtual brushes. In *SIGGRAPH.* 461–468.
[11] Fabrizio Bianchi. 2018. Coolors. https://coolors.co.
[12] Anthony M Blatner, James A Ferwerda, Benjamin A Darling, and Reynold J Bailey. 2011. TangiPaint: A Tangible Digital Painting System. *Color Imaging Conference* (2011), 102–107.
[13] Bohemian B.V. 2018. Sketch. https://www.sketchapp.com.
[14] Canva. 2018. Color Palette Generator. https://www.canva.com/color-palette/.
[15] Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based photo recoloring. *ACM Trans. on Graphics (TOG)* 34, 4 (2015), 139.
[16] Shaun Chapman. 2018. 0to255. http://www.0to255.com.
[17] CharlieMarieTV. 2017. Colour palette management tutorial. https://www.youtube.com/watch?v=zl_7yHh2PSc.
[18] Xiaowu Chen, Dongqing Zou, Jianwei Li, Xiaochun Cao, Qinping Zhao, and Hao Zhang. 2014. Sparse dictionary learning for edit propagation of high-resolution images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition.* 2854–2861.
[19] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 4 (2014), 21.
[20] Alex Clem. 2018. Everything You Need to Know About Gradients in Design. https://www.shutterstock.com/blog/complete-guide-gradients-designs.
[21] ColorBlender. 2018. ColorBlender. http://colorblender.com.
[22] ColorHexa. 2018. ColorHexa. https://www.colorhexa.com/color-gradient.
[23] Colormind.io. 2018. Colormind. http://colormind.io/.
[24] colorspire. 2018. Colorspire. https://www.colorspire.com/.
[25] COLOURlovers. 2018. COLOURlovers CC. http://www.colourlovers.com.
[26] Colrd. 2018. Colrd. http://colrd.com/create/palette/.
[27] 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.
[28] Commision Internationale de l'Éclaraige. 2001. *Improvement to industrial colour-difference evaluation.* Technical Report 142-2001. Central Bureau of the CIE.
[29] Andrew M Derrington, John Krauskopf, and Peter Lennie. 1984. Chromatic mechanisms in lateral geniculate nucleus of macaque. *The Journal of Physiology* 357, 1 (1984), 241–265.
[30] Sara Duane-Gladden. 2018. 10 Handy Tips To Keep Your Color Swatches and Palettes Organized. https://www.hongkiat.com/blog/color-swatches-organized.
[31] Marc Edwards. 2017. Taming Advanced Color Palettes In Photoshop, Sketch And Affinity Designer. https://www.smashingmagazine.com/2017/07/advanced-color-palettes-photoshop-sketch-affinity-designer/. Accessed: 2018-08-19.
[32] Eggradients. 2018. Eggradients. https://www.eggradients.com.
[33] Leatrice Eiseman. 2017. *The Complete Color Harmony, Pantone Edition: Expert Color Information for Professional Results.* Rockport Publishers.
[34] Andrew J Elliot and Markus A Maier. 2014. Color psychology: Effects of perceiving color on psychological functioning in humans. *Annual review of psychology* 65 (2014), 95–120.
[35] Indrashish Ghosh. 2018. uiGradients. https://uigradients.com.
[36] Google. 2018. Material Design Color Tool. https://material.io/tools/color.
[37] hailpixel. 2018. Colordot. https://color.hailpixel.com.
[38] Harry G. Hecht. 1983. A Comparison of the Kubelka-Munk, Rozenberg, and Pitts-Giovanelli Methods of Analysis of Diffuse Reflectance for Several Model Systems. *Applied Spectroscopy* 37, 4 (1983), 315–403.
[39] Jeffrey Heer and Maureen Stone. 2012. Color Naming Models for Color Selection, Image Editing and Palette Design. In *CHI.* 1007–1016. http://vis.stanford.edu/papers/color-naming-models
[40] 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. https://doi.org/10.1002/col.21762
[41] HueSnap. 2018. HueSnap. https://www.huesnap.com.
[42] IDEA. 2018. ColorRotate. http://mobile.colorotate.org.
[43] Itmeo. 2018. Web Gradients. https://webgradients.com/.
[44] Ghita Jalal, Nolwenn Maudet, and Wendy E. Mackay. 2015. Color Portraits: From Color Picking to Interacting with Color. *CHI* (2015), 4207–4216.
[45] Joseph Angelo Todaro. 2017. How I Organize Colors for a UI Design Project. https://www.youtube.com/watch?v=9S8LGdpNh4Q.
[46] Jim Krause. 2017. *Color Index XL: More than 1,100 New Palettes with CMYK and RGB Formulas for Designers and Artists.* Watson-Guptill.
[47] Sharon Lin and Pat Hanrahan. 2013. Modeling how people extract color themes from images. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 3101–3110.
[48] Sharon Lin, Daniel Ritchie, Matthew Fisher, and Pat Hanrahan. 2013. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Trans. on Graphics (TOG)* 32, 4 (2013), 37.
[49] Jingwan Lu, Stephen DiVerdi, Willa A Chen, Connelly Barnes, and Adam Finkelstein. 2014. RealPigment: Paint compositing by example. In *NPAR.* 21–30.
[50] Donald IA MacLeod and Robert M Boynton. 1979. Chromaticity diagram showing cone excitation by stimuli of equal luminance. *JOSA* 69, 8 (1979), 1183–1186.
[51] Barbara J Meier, Anne Morgan Spalter, and David B Karelitz. 2004. Interactive color palette tools. *IEEE Comp. Graph. App.* 24, 3 (2004), 64–72.
[52] Nicolas Mellado, David Vanderhaeghe, Charlotte Hoarau, Sidonie Christophe, Mathieu Brédif, and Loic Barthe. 2017. Constrained palette-space exploration. *ACM Trans. on Graphics (TOG)* 36, 4 (2017), 60.
[53] Mindsparkle Mag. 2018. 2018 Design Trends. https://www.behance.net/gallery/60273889/2018-Design-Trends. Accessed: 2018-08-19.
[54] Jeanne Nakamura and Mihaly Csikszentmihalyi. 2014. The concept of flow. In *Flow and the foundations of positive psychology.* Springer, 239–263.

[55] 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.
[56] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2011. Color compatibility from large datasets. *ACM Trans. on Graphics (TOG)* 30, 4 (2011), 63.
[57] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Collaborative filtering of color aesthetics. *Computational Aesthetics* (2014), 33–40.
[58] Our Own Thing. 2018. Grad Pad. http://ourownthing.co.uk/gradpad.html.
[59] Paletton. 2018. Paletton. http://paletton.com.
[60] Paper Fifty Three Inc. 2017. Paper by FiftyThree - Sketch, Diagram, Take Notes. https://itunes.apple.com/us/app/paper-by-fiftythree-sketch/id506003812?mt=8.
[61] PerBang.dk Color. 2018. RGB Chart & Multi Tool. http://www.perbang.dk/rgb/.
[62] PerBang.dk Color. 2018. RGB Color Gradient Maker. http://www.perbang.dk/rgbgradient/.
[63] Huy Q Phan, Hongbo Fu, and Antoni B Chan. 2018. Color Orchestra: Ordering Color Palettes for Interpolation and Prediction. *IEEE transactions on visualization and computer graphics* 24, 6 (2018), 1942–1955.
[64] Port80.biz. 2018. ColorExplorer. http://colorexplorer.com.
[65] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design principles for tools to support creative thinking. In *NSF Workshop Report on Creativity Support Tools*. Citeseer, 25–36.
[66] Rikku. 2017. Color Space. https://mycolor.space/.
[67] Ryan McCready. 2018. 8 New Graphic Design Trends That Will Take Over 2018. https://venngage.com/blog/graphic-design-trends/. Accessed: 2018-08-19.
[68] Gal Shir. 2018. Color Hunt. https://colorhunt.co.
[69] Ben Shneiderman. 2007. Creativity support tools: Accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32.
[70] Maria Shugrina, Jingwan Lu, and Stephen Diverdi. 2017. Playful palette: an interactive parametric color mixer for artists. *ACM Trans. on Graphics (TOG)* 36, 4 (2017), 61.
[71] Alvy Ray Smith. 1978. Color gamut transform pairs. In *SIGGRAPH*. 12–19.
[72] Alvy Ray Smith and Eric Ray Lyons. 1996. HWB—A More Intuitive Hue-Based Color Model. *Journal of Graphics Tools* 1, 1 (Jan. 1996), 3–17.
[73] KyoungHee 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 *UIST Adjunct*. 95–96.
[74] Ian Spalter. 2016. Designing a New Look for Instagram, Inspired by the Community. https://medium.com/@ianspalter/84530eb355e3. Accessed: 2018-08-19.
[75] Unfold. 2018. Grabient. https://www.grabient.com/.
[76] UX Planet. 2018. Why Gradients are back to rule in 2018? https://uxplanet.org/why-gradients-are-back-to-rule-in-2018-8b36711c335f. Accessed: 2018-08-19.
[77] Peter Vandoren, Tom Van Laerhoven, Luc Claesen, Johannes Taelman, Chris Raymaekers, and Frank Van Reeth. 2008. IntuPaint: Bridging the gap between physical and digital painting. *TABLETOP* (2008), 65–72.
[78] Lauren Wager. 2018. *Color Collective's Palette Perfect: Color Combinations Inspired by Fashion, Art and Style*. Promopress.
[79] Baoyuan Wang, Yizhou Yu, Tien-Tsin Wong, Chun Chen, and Ying-Qing Xu. 2010. Data-driven image color theme enhancement. In *ACM Trans. on Graphics (TOG)*, Vol. 29. ACM, 146.
[80] Webkul Software. 2018. Cool Hue. https://webkul.github.io/coolhue/.
[81] Jane Webster, Linda Klebe Trevino, and Lisa Ryan. 1993. The dimensionality and correlates of flow in human-computer interactions. *Computers in human behavior* 9, 4 (1993), 411–426.
[82] 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. https://doi.org/10.1111/j.1467-8659.2008.01203.x
[83] Terry Winograd. 1996. *Bringing Design to Software*. Addison-Wesley, Chapter Reflective Conversation with Materials.
[84] X-Rite Incorporated. 2018. Color Munki. http://www.colormunki.com/pantone.

A PILOT QUESTIONNAIRE

For completeness, we list some of the questions from the pilot questionnaire we distributed to design professionals to help guide our initial design (§3):

- What is important to you in tools (digital or physical) for working with color? What frustrates and delights you in tools that are available to you?
- Are there specific parts of your color workflow that are particularly easy or difficult to manage with current tools, digital or physical?
- What color relationships do you consider? E.g, describe how you use contrasting or coordinated colors. What tools/techniques do you use to help visualize or manage these color relationships?
- Do you use gradients/color blends? If so, how do you create them? How do you edit gradations of colors if your initial palette changes?
- If one of the colors in your design needs adjustment, how do you tweak it? What are your favorite tools / methods for that?
- What do you like and dislike about available tools for adjusting a single color?