

# Color Picking: The Initial 20s

JAN KOENDERINK, Department of Brain & Cognition, Laboratory of Experimental Psychology, University of Leuven (KU Leuven)

ANDREA VAN DOORN, Experimental Psychology, University of Utrecht

VEBJØRN EKROLL, Department of Brain & Cognition, Laboratory of Experimental Psychology, University of Leuven (KU Leuven)

Color pickers are widely used in all kinds of display applications. They vary greatly in their utility, depending on user expertise. We focus on nonprofessional, occasional users. Such users may spend from a few seconds up to a few minutes to select a color. Yet, typically they reach final accuracy within the initial 20s. Additional effort leads to random walks in the neighborhood of the target. We explore the efficaciousness of five generic color pickers, analyzing the results in terms of generic user interface properties. There is a major dichotomy between three-slider interfaces, and those that offer some form of 2D selectivity. The accuracy in RGB coordinates is about one-tenth to one-twentieth of the full scale (often 0–255 in R, G, and B), whereas a little over 100 hues are resolved. The most efficient color picker, which is presently rarely used in popular applications, is much more efficient than the worst one. We speculate that this derives from a closer match to the user's internal representation of color space. The study results in explicit recommendations for the implementation of user-friendly and efficient color tools.

Categories and Subject Descriptors: H.5.2 [User Interfaces]: Ergonomics—*Evaluation / methodology*

General Terms: Design, Human Factors, Performance

Additional Key Words and Phrases: Color pickers

## ACM Reference Format:

Jan Koenderink, Andrea van Doorn, and Vebjørn Ekroll. 2016. Color picking: The initial 20s. *ACM Trans. Appl. Percept.* 13, 3, Article 13 (April 2016), 25 pages.

DOI: <http://dx.doi.org/10.1145/2883613>

## 1. INTRODUCTION

“Color pickers” are used in numerous common applications [Foley et al. 2005]. They are typically used infrequently, sometimes even on a one-time basis and often by naïve users. Experienced users have little problem with almost any color picker design, as long as it is familiar to them. For such users the interface design hardly matters. In contradistinction, first-time, infrequent, or naïve users often encounter problems, and the design greatly matters. For instance, few naïve users manage well with the conventional, and historically prior, red, green, and blue linear slider design [Smith 1978]. Moreover,

This work was supported by the Methusalem program by the Flemish Government (METH/08/02), awarded to Johan Wagemans. Authors' addresses: J. Koenderink and V. Ekroll, Department of Brain & Cognition, Laboratory of Experimental Psychology, University of Leuven (KU Leuven), Tiensestraat 102, box 3711, Belgium; emails: [koenderinkjan@gmail.com](mailto:koenderinkjan@gmail.com), [vebjorn.ekroll@kuleuven.be](mailto:vebjorn.ekroll@kuleuven.be); A. van Doorn, Experimental Psychology, University of Utrecht, Heidelberglaan 2, 3584 CS Utrecht, The Netherlands; email: [andrea.vandoorn@telfort.nl](mailto:andrea.vandoorn@telfort.nl).

Permission to make digital or hard copies of part or all 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 show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 2016 ACM 1544-3558/2016/04-ART13 \$15.00

DOI: <http://dx.doi.org/10.1145/2883613>

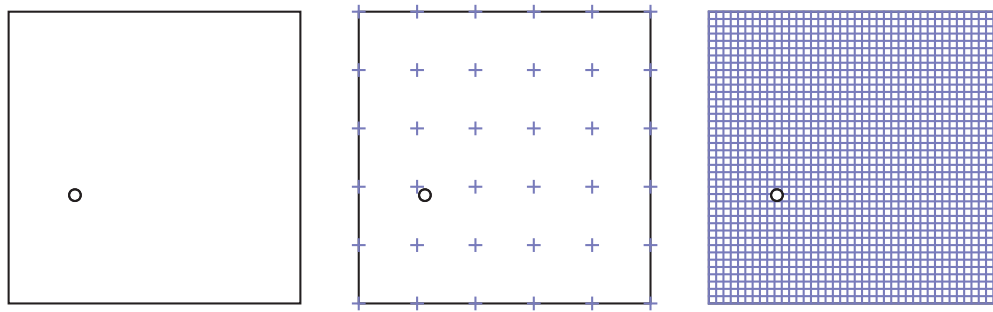


Fig. 1. Example of a difficulty induced by the interface graphics. In many color pickers one needs to define a pair of parameters by indicating a location in a square or circle (left). Adding some anchors greatly helps (center); overdoing it defies the purpose (right). When the area is not homogeneous (as in most radial parameterization the center is) things become even more complicated.

as we established in this study, naïve users thoroughly dislike such an interface. Yet, perhaps surprisingly, this design comes out favorably in prior research [Schwarz et al. 1987; Douglas and Kirkpatrick 1999]. We conclude that one should not only consider the outcome, but also the way of establishing it.

In this article, we consider the relative merits of various color picker designs on the basis of many trials of a group of naïve, first-time users. We relate the results to generic interface design principles, as well as the naïve user’s internal model of the space of RGB colors. We conclude with specific recommendations. These differ from conventional implementation choices in widely used applications.

A Graphical User Interface (GUI) [Foley et al. 2005] consists of a smallish number of graphical elements, and a set of corresponding interactions associated with them. Apart from the GUI elements proper, there is an element called **RESULT**, which is also presented graphically, but has no specific interaction associated with it. There is also an object called **GOAL**, which may, or may not have a graphical representation.

In many applications the **GOAL** will be a mere mental object, for example, a painter may have some hue in mind, and mix pigments to materialize it. In other cases the **GOAL** may be explicitly present. In this article, we only consider this latter case. The resulting dab of color (the “**RESULT**”) may, or may not, approximate the **GOAL**. In virtually any application **RESULT** and **GOAL** cannot be directly compared, but are separated in space-time. Some form of “mental transfer” is required.

“Color picking” taxes dexterity, visual recognition, and working memory. Bottlenecks may appear in any of these departments. Various definitions of the task, algorithmic structure of the color tool, its graphical layout (Figure 1), and interactive elements of the interface, all enter as factors. The user’s mental model of RGB-color space is likely to play an important role.

The user interacts with the GUI occasionally watching the **GOAL** and continually keeping it in mind. The user watches the **RESULT** change in response to the interactions. At some point the user will either be satisfied that the **RESULT** matches the **GOAL**, or will give up, in which case some frustration results. The efficaciousness of the GUI can be assessed in terms of the time it takes to approximately reach the **GOAL**, and the accuracy of the match.

The efficaciousness is a property of the GUI, the user, and the **GOAL**. It often makes sense to generalize over **GOALS** and/or over users. In this article, we consider only naïve, first-time users, and a uniform distribution of RGB colors as **GOAL** space. This closely approximates the daily use of “color pickers” in a variety of applications involving graphics or photographic images.

## 2. METHODS

The experiment was implemented in Object-C [Cox 1983] (Object-C was adopted by Apples Cocoa framework [Hillegass 2008]) on Apple computers under OS X (a form of UNIX). We did not use the standard Cocoa GUI elements (sliders, buttons, cursors, and so forth), but implemented all elements from basic graphical primitives (disks and rectangles). Thus, our GUIs look like universal, generic GUIs, in no obvious way related to any familiar one. This is important, because our observers have a variety (Apple OS X Aqua, Microsoft Windows, Java swing, tkinter, Qt, ...) of daily experiences. One thing beyond our control is that most users will have Adobe color picker experiences (Photoshop [Brundage 2012] for their home pictures), or Apple color picker experiences (one of the generic OS X tools). This is likely to bias their preferences, and abilities. Moreover, a few occasional programmers will perhaps have acquired a feeling for the RGB interface.

Observers were 11 graduate students and postdocs from the Department of Experimental Psychology of Leuven University (KU Leuven). This amply suffices to make our various conceptual points, but is hardly an unbiased population sample.

We used RGB colors on a DELL U2410f monitor with standard settings (gamma 2.2 [Poynton 1998], white point 5700°K [Wyszecki and Stiles 1928]), a 1920 × 1200 pixels LCD screen, in a darkened room. The screen was calibrated with a Minolta CS-1000 spectroradiometer and linearized with the SuperCal application (<http://www.bergdesign.com/supercal/>). The xyY-colorimetric coordinates of the “primaries” were red {0.6049, 0.3416, 94.3}, green {0.3295, 0.5750, 272.7}, and blue {0.1531, 0.1382, 71.1}.

The reason that the color temperature of the white was set to 5700°K, and that we used gamma 2.2 was because this was not a colorimetric study. In contradistinction, we attempted to approximate typical use. The viewing distance was 78cm, and the screen was 51.7cm wide. The stimulus filled the width of the screen.

The screen was divided equally into a left and a right part. The right part contained the color picker, and the left part the GOAL and the RESULT. Both were implemented as colored disks. The disks were presented on a random mosaic background of which color statistics mimicked that of natural object colors [Koenderink 2010b]. The disks were filled with uniform color over which we superimposed a cobweb line structure in very thin line (see Figure 2). The latter served to render a “material,” as opposed to “ethereal,” appearance of GOAL and RESULT. The background was supplied to “anchor” the RGB colors by generic context. For instance, without such “anchoring” the white point, and thus RGB space, is undefined. This closely approximates “normal use,” whereas still presenting a controlled laboratory setting.

The observers interacted with the GUI with a single-button mouse, using dragging and clicking actions. A drag changes a parameter continuously, whereas a click may change it discontinuously. All user interactions were recorded, together with their time stamps. Observers were instructed to carry on until they were satisfied with their match, or sufficiently frustrated to give up. Of course, the latter line of action was not encouraged. However, it was suggested that it might not be productive to carry on for more than about 2 minutes. This yields considerable data volumes, allowing arbitrarily detailed analysis of the interaction. Observers were not aware of the data collection, they merely attempted to produce the GOAL. Thus a session mimics typical first-time use.

### 2.1 The Color Pickers Used in This Study

We implemented five color pickers, roughly modeled after conventional designs (Figure 3). The limitation to just five was dictated by pragmatic constraints. Consequently, the choice is somewhat arbitrary. Although our implementations were inspired by well known examples, we did in no way attempt to reproduce the originals. Our objective is not to show that any type “is better” than any other. As said

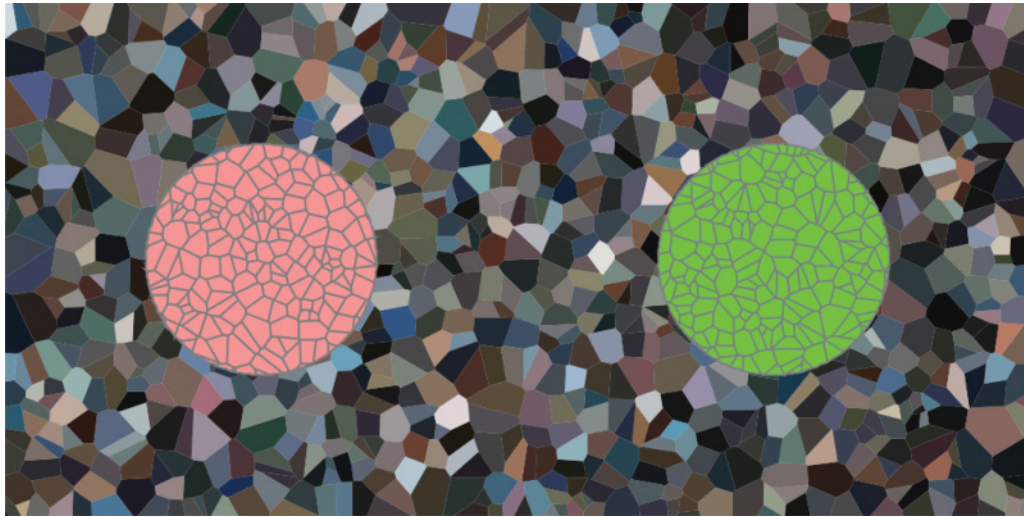


Fig. 2. Two patches (test and goal) on a neutral background. The background is generated by a random natural gamut algorithm; the text patches are uniform, but overlaid with a pattern of average gray grout. This makes the patches appear “material” and sets them off against all other colors, thus avoiding systematic contrast biases. (This image has been edited in various ways so as to appear reasonable in print.)

earlier, the experienced user will be happy with any of these, and is even likely to be ready to defend it against all comers. Most people get hooked on familiar tools. We label the instances I, II, III, IV, and V.

*Color Picker I.* This GUI implements a common design, similar to those often encountered in some form in Adobe applications. It has a square mouse interaction region, and a single linear slider. It is hard on the naïve user. The top edge of the square is not responsive, and the left and right sides are topologically merged, and thus have to be mentally identified by the user. No anchor points are provided. The (unfortunate) square is probably inspired by the conventional “rectangle” of common graphics libraries, reflecting common window systems.

*Color Picker II.* This GUI implements another common design, similar to those often encountered in some form in Apple applications. It has a circular disk mouse interaction region, and a single linear slider. It is also hard on the naïve user. No anchor points are provided. This is especially awkward because the center is special: it is a singular point for hue. In various implementations the disk becomes uniform at one extreme slider setting, but we did not include that “feature.” The circular disc is probably a distant echo of Newton’s “color circle.”

*Color Picker III.* This GUI implements a design that is not often encountered in graphics applications, but is based on common convictions of the experimental phenomenology of color, the “opponent color” parameterization originally suggested by Hering [1964]. Perhaps unfortunately, and perhaps surprising to vision scientists, it is very hard on naïve users. One obvious reason is the nonlinear interaction between the GUI elements, which tend to confuse users. The interface elements are three linear sliders: one representing a unipolar (i.e.,  $[0,1]$ ) domain, and two others representing bipolar (i.e.,  $[-1, +1]$ ) domains.

*Color Picker IV.* This GUI is based on the structure of color atlases suggested (and implemented as a hand-painted book in 1917) by Ostwald [1917, 1919]. Its GUI elements are an annulus, and a solid triangle. It manages to handle most of the aforementioned difficulties in a natural way. It is a GUI that first-time users spontaneously like. A closely related CG system is HWB, proposed by Smith

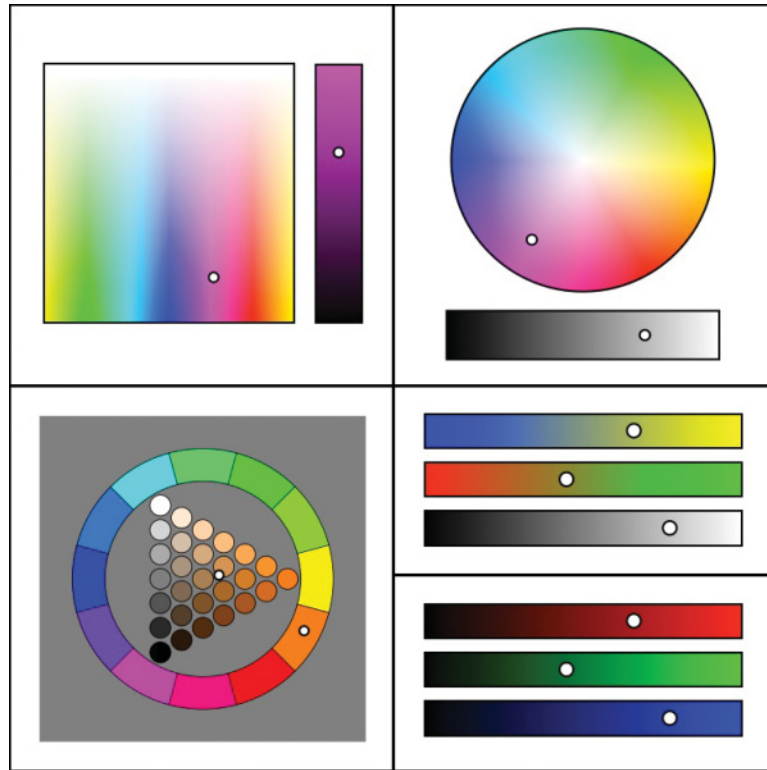


Fig. 3. Color picker graphics. Top left color picker I, top right color picker II, bottom left color picker IV, and bottom right color pickers III (top), and V (bottom).

and Lyons [1996] as a replacement of HSV and HSL. Smith as the originator [Smith 1978] of HSV and HSL, calls these latter systems “flawed,” and recommends HWB as especially friendly to naïve users.

Notice the discrete samples in the triangle. This implements the principle of “anchoring” illustrated in Figure 1.

*Color Picker V.* This GUI implements the basic triplet of linear RGB sliders, often preferred by professional designers. All sliders represent identical unipolar domains. In the printing world CMY sliders would be the preferred choice, but the abstract GUI is identical. (One might add an additional slider for the density of the black plate though.) It is a difficult GUI to use for naïve users, who can hardly guess that one needs to decrease both R and G by the same amount, thus using two interaction domains simultaneously, in order to make a white RESULT bluish, and so forth.

We explain the particular parameterizations used in these implementations, as well as their graphical representations, in the Appendix.

Notice that in all cases the momentary setting fully determines the color. Thus none of the color pickers suffers from problems due to noncommutativity. Of course, the users will likely suffer from hysteresis effects. These will influence the eventual accuracy and time to target; our methods capture that as indeed desired.



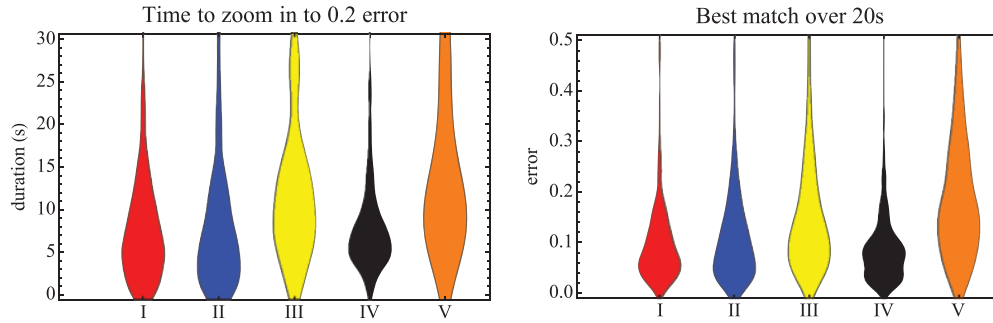


Fig. 4. Distribution plots of the main overall characteristics of the five interfaces. The p.d.f.'s are over 20 trials, and 11 observers.

Table I. At Left Quartiles of the Full Duration, at Right Quartiles of the Time to Criterion, all Expressed in Seconds. Values are Tabulated for Color Pickers I–V

	Q25	Q50	Q75		Q25	Q50	Q75
I	10.6s	58.4s	140.7s	I	3.6s	6.5s	11.2s
II	5.6s	47.8s	109.7s	II	2.7s	6.2s	11.9s
III	7.0s	51.7s	145.9s	III	6.5s	11.0s	17.3s
IV	13.3s	54.2s	108.7s	IV	4.5s	6.7s	9.8s
V	7.6s	63.1s	139.3s	V	7.4s	11.8s	24.5s

### 3. EXPERIMENT

The actions of an observer depend upon the initial condition, that is, the **RESULT** (a misnomer in the initial case, because forced) set to some specific value, and the **GOAL**. Both are points in RGB space, which may be formally represented by the unit RGB cube. For the initial **RESULT**-**GOAL** pairs we used antipodal points from a dodecahedron inscribed in the RGB cube, yielding 20 pairs. These were visited in random sequence. We used the same random sequence in all cases in order to be able to compare observers and GUIs on equal terms. The results are numerous orbits in RGB space, brought about through the user interactions. This huge dataset may be mined in numerous ways; our present analysis represents only the tip of the iceberg.

In this section, we discuss some of the global properties of the data. A more in-depth, perhaps slightly speculative, analysis is offered in the next section.

Users behave in very different ways (Figure 4). This is already obvious from the time they spend on reaching a match or giving up, which varies from 5s to almost 5 min. Quartile ranges and median (table, left) show that the differences are perhaps mainly due to the observers, and not so much on the particular GUI. Of course, this is only to be expected. If one is interested in timing, a much better measure is to see how much time is needed to “zoom in” to an approximate neighborhood of the **GOAL**. One has to set some arbitrary limit here, we take a distance (Euclidean distance in the RGB cube) of 0.2 from the goal, which is two or three times the final accuracy. Here we find a clear dichotomy between interfaces {III, V}, and {I, II, IV}, as is evident from the quartiles {Q25, Q50, Q75} (at left the full duration, at right the time to criterion; all units in seconds) (Table I).

The time to criterion clearly reveals the differences between the GUIs, whereas the full durations are hardly informative. (This is something to keep in mind when comparing related literature data.) For interfaces I, II, and IV the median time is in the 6–7s range, whereas for interfaces III and V it is about 11–12s, almost double the former values (see Figure 4 (left)).

Table II. At Left the Deviation of the Best Match from the Goal after 20s, and at Right the Ultimate Precision. These Values are Distances in the RGB-Cube. The Accuracy of the Result is Best Understood by Comparing them to the Length of an Edge of the Cube, which is the Maximum Value (1) of Any Color Coordinate (R, G, or B)

	Q25	Q50	Q75		Q25	Q50	Q75
I	0.04	0.07	0.10	I	0.05	0.08	0.13
II	0.04	0.07	0.11	II	0.04	0.09	0.15
III	0.05	0.19	0.17	III	0.07	0.12	0.21
IV	0.04	0.06	0.10	IV	0.04	0.07	0.11
V	0.06	0.09	0.16	V	0.10	0.16	0.26

Table III. Quartiles of the FOM. The FOM is Defined as the Reciprocal of the Product of the Median Zooming-in Time (in Seconds) and the Error of the Median Best Match (Fraction of the Maximum Coordinate Value of 1) Over 20s. Thus an Accuracy of 1% Obtained within 10s—the Kind of Result One Might Perhaps Intuitively Hope for—would Imply a FOM of  $1/(0.01 \times 10) = 10$ . None of the Interfaces Lets One Reach That

	Q25	Q50	Q75
I	3.6	4.2	5.6
II	2.7	3.5	5.8
III	0.8	1.7	2.1
IV	3.7	4.7	5.3
V	0.7	1.2	2.3

A similar pattern is found for the accuracy of the best matches. Table II (left) shows the best match after 20s, and Table II (right) the ultimate precision (values are fractions of the maximum value of 1).

The precision over the initial 20s differentiates much better between the GUIs than the ultimate precision does. In the latter case the real differences are masked because the errant behavior in the final stage may well approach the GOAL by accident. That is why these values are rather even, and uninformative. (Again, this is something to keep in mind when comparing related literature data.) The most obvious—and important—differences are again obtained from an analysis of the initial 20s (Figure 4 (right)). The precision is 0.06–0.09 for interfaces I, II, IV, and V, whereas it is 0.19 for interface III. Evidently, interfaces {I, II, IV, V} are far more efficient than interface III. The value for interface III is about twice that for interfaces {I, II, IV, V}.

Notice the best-case precision is still much worse than the just noticeable differences found in the classical split field discrimination task [MacAdam 1943]. This is hardly surprising, since ours is not a split-field matching task (see discussion).

If an overall “winner” is desired, one might consider a Figure Of Merit (FOM) defined as the reciprocal of the product of the median zooming-in time and the error of the median best match over 20s (notice that higher is better; see Table III).

As expected, the aforementioned dichotomy is evident. The FOM’s for interfaces I, II, and IV are 3.5–4.7, whereas those for interfaces III and V are 1.2–1.7. (The physical dimension of the FOM is reciprocal

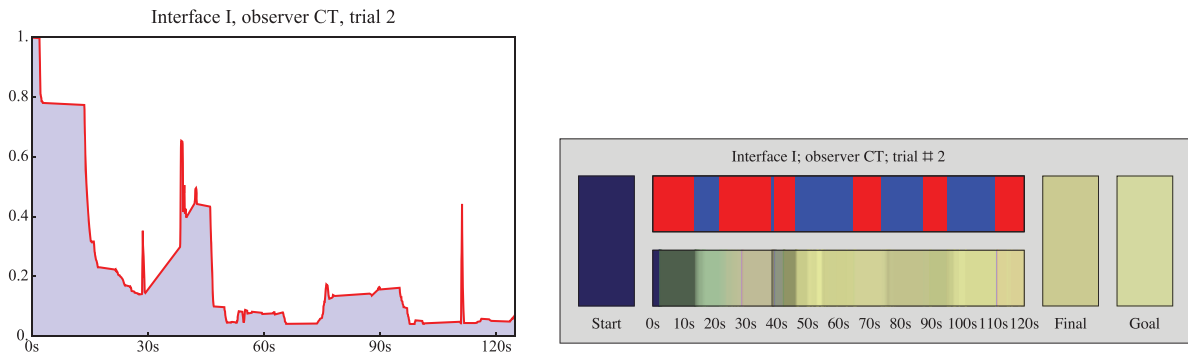


Fig. 5. The settings of observer CT for trial #2, interface I. The graph at left shows the mismatch (a number between zero (perfect match) and 1 (1 is the full r, g, b coordinate range, the worst possible error is  $\sqrt{3} = 1.732 \dots$ )) as a function of time (in seconds). It obviously decreases, but not monotonically: notice the spikes. At right we show the RESULT (bottom trace), and the GUI elements in current use (blue the square, red the slider). The spikes happen when the user has to move the cursor over the width of the square in order to arrive at (almost) the “same” location. (The initial “RESULT” is labeled “Start,” the GOAL is labeled “Goal,” and the final RESULT is labeled “Final.”)

time, but this is irrelevant here because we only consider ratios.) Thus, interface V is evidently the worst, whereas interface IV is the best, albeit by a minor (nonsignificant) margin. The ratio between the best and the worst FOMs is fourfold, which is very large. Even a cursory look at the interquartile ranges shows that no significance test is called for here.

The numbers mentioned here essentially suffice to document the huge gap between interfaces {I, II, IV} (basically okay), and {III, V} (no doubt bad). In the next section, we analyze these differences in more detail, and speculate about likely “reasons.”

It is perhaps interesting to mention that observers were almost unanimous in hating interfaces III and V (no evident overall loser), and considered I, II, and IV okay, with a decided preference for IV. This is somewhat remarkable, because most had regular exposure to either I (various Adobe applications), or II (various Apple applications), whereas no one had previously encountered IV (we know only the Corel Painter application (Zimmer 2012), which comes close, but is not identical).

#### 4. ANALYSIS

A closer study of the ways observers reached (or failed to reach) the target, provides many useful insights into GUI design. We discuss a few striking examples. The technical terms pertaining to RGB colorimetry and the formal structure of the color pickers, are explained in the Appendix.

Perhaps the worst design glitch in interface I is that the main GUI element, the square, does not reflect the basic topology of the hue domain. This leads to frequent unfortunate happenings. Consider trial #2 of observer CT (Figure 5). Notice the spikes at 30s, 40s, and 115s. Even near the very end the mismatch suddenly exploded, albeit only temporarily. What happened is that the observer had to move the cursor over the whole interface in order to arrive at a nearby point!

A few typical features of a graph such as Figure 5 (left) are perhaps worth pointing out. The graph is roughly monotonically decreasing, although with frequent interruptions. Eventually it enters a regime of random fluctuation. The initial, mainly decreasing part, shows a (usually small) number of plateaus. Here the user is “caught” in a certain state, from which it is hard to escape. Occasionally such a plateau may be above an earlier achievement. The user has to do some random poking in order to obtain a “feel” for the interface.

In Figure 6, we illustrate an effect common to interfaces I and II. Notice the glitch in the RESULT at 5s.



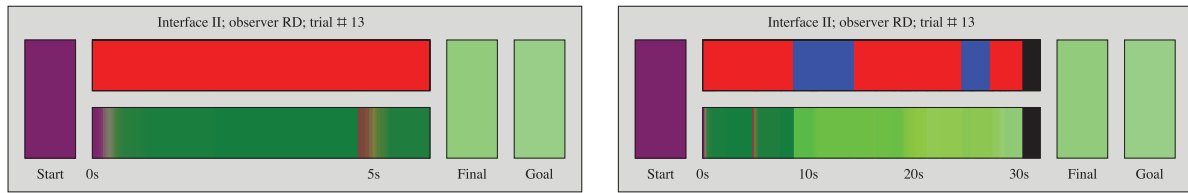


Fig. 6. Both graphs interface II, observer RD, trial #13, shown at different time scales. Notice the glitch at 5s. Here red indicated actuation of the disk GUI element, and blue that of the slider.

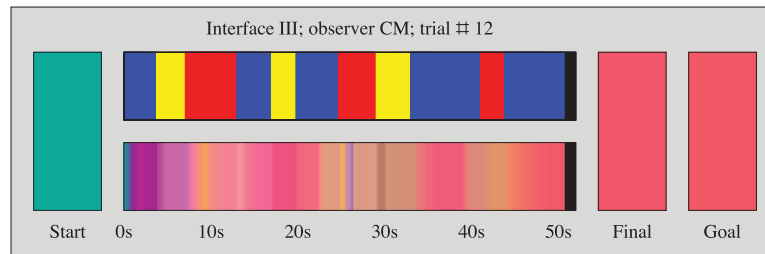


Fig. 7. Interface III, observer CM, trial #12. The observer is “hunting” in order to get in the neighborhood of the GOAL, without much success. The final match, after about a minute, was already reached in less than half this time. “Corrections” only led further away from the GOAL. (Here red signified the YB slider, blue the RG slider, and yellow the KW slider.)

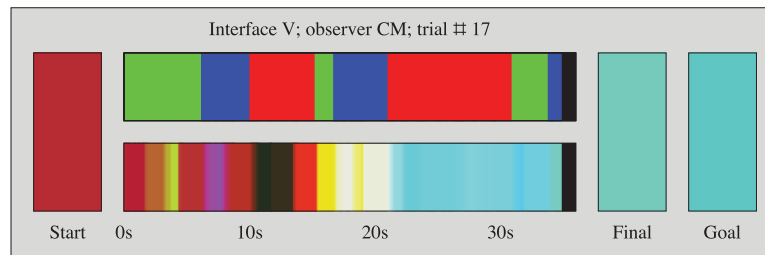


Fig. 8. Interface V, observer CM, trial #17. The observer is “hunting” in order to get in the neighborhood of the GOAL. The fluctuations appear to die out a little after about 20s though. Before that one has an errant orbit traversing large parts of RGB space. Here red signified the R slider, blue the B slider, and green the G slider.

What happened? Well, the observer moved past the center and saw the hue change to its complement. The reason is that the center is not indicated by some anchor point. Without anchoring, the location in extended GUI elements is very indeterminate.

The major problems with interface III are the nonlinear interactions between the GUI elements. This confuses users, and leads to “hunting” behavior. An example is shown in Figure 7. It is much the same for interface V (Figure 8). Observers are evidently puzzled by the interactions.

Interface IV is experienced as “nice” by all observers (Figure 9). Notice that they use fewer switches of GUI elements, and rather monotonically approach the target, without obvious glitches.

Of course, there are many ways to analyze these data. To plot the mismatch between GOAL and result as a function of time, yields one obvious, but very summary account. It is somewhat nice because it enables one to show the achievements of all observers in one graph. From the example (Figure 10) one sees that nothing is gained by extending the interaction beyond about 20 or 30s.

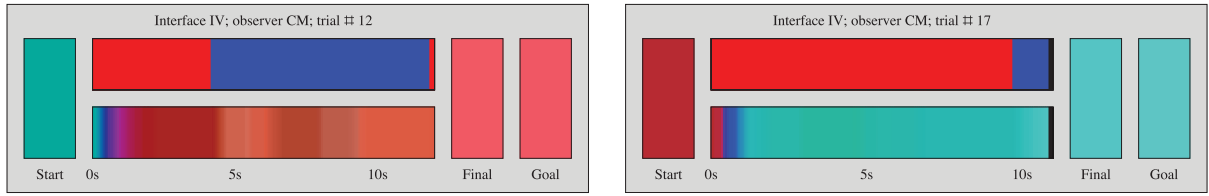


Fig. 9. Interface IV and observer CM on trial #12 (compare Figure 7) and #17 (compare Figure 8). When comparing these figures mind the time scales! For interface IV, red stands for the color circle, and blue for the triangular GUI element.

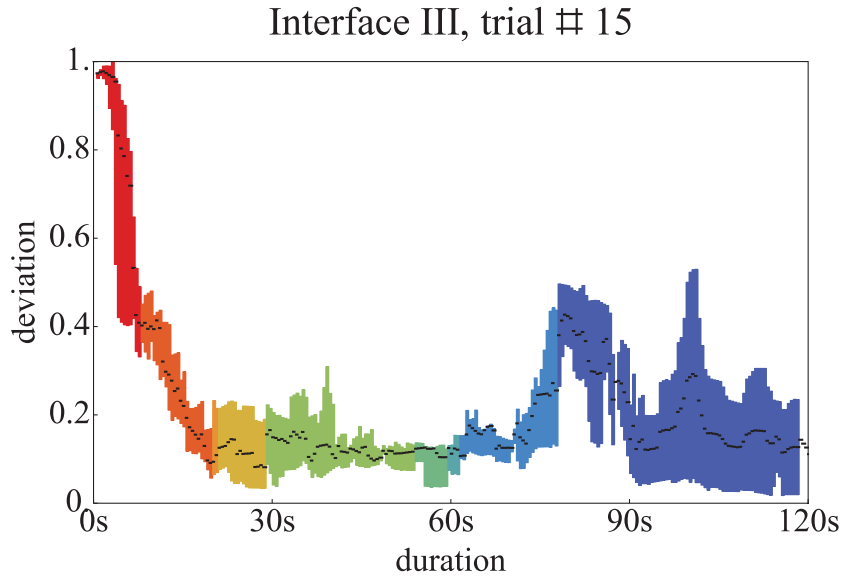


Fig. 10. The mismatch between RESULT and GOAL as a function of time, for all observers. The mismatch is defined as the Euclidean distance in the RGB unit cube. Notice that the various observers stop at different times. We use the hue to indicate this; a hue shift indicates a change in the number of active observers. The black points show the median, and the colored bars the IQR.

The final (lowest) plateau is about 0.1, thus the precision is much lower than the Just Noticeable Difference (JND) for the usual abutting half-fields. Indeed, one immediately notices the variations in the final achievements of the observers (Figure 10).

It is of some interest to consider the behavior in subspaces that are likely to be of phenomenological relevance. A good example is hue (Figures 11 and 12). We immediately notice major differences between the various interfaces. We fitted a low order polynomial expression to the time course of the hue difference from the goal. Order three proves to yield a natural “smoothed” representation. The deviations from the fit evidently show the irregularity in the approach of the goal. We simply computed the variance over a uniformly sampled temporal interval as a measure of wriggleness. This immediately identifies interfaces III and V as awkward. Compare the quartile ranges of the wriggleness in Table IV.

The differences between interfaces I, II, and IV is immediately obvious from the wriggleness measures. The approach to the GOAL is smoothest (median wriggleness 0.03) for GUI IV, and most contorted (median wriggleness 0.14, about five times more) for GUI V.



Fig. 11. The final results, for all 11 observers, for the various trials of interface V. The GOAL is shown as the disk at left. Notice that the spread in the results is immediately apparent if one sees all results simultaneously. Interface V yields the worst-case performance, in spite of the fact that it is perhaps the most commonly used.

Table IV. Quartiles of the “Wiggleness.” The Wiggleness is Defined as the Variance Over a Uniformly Sampled Temporal Interval. The Values are Expressed as Fractions of the Maximum Coordinate Value (1)

	Q25	Q50	Q75
I	0.03	0.05	0.09
II	0.02	0.04	0.08
III	0.04	0.08	0.15
IV	0.01	0.03	0.06
V	0.07	0.14	0.26

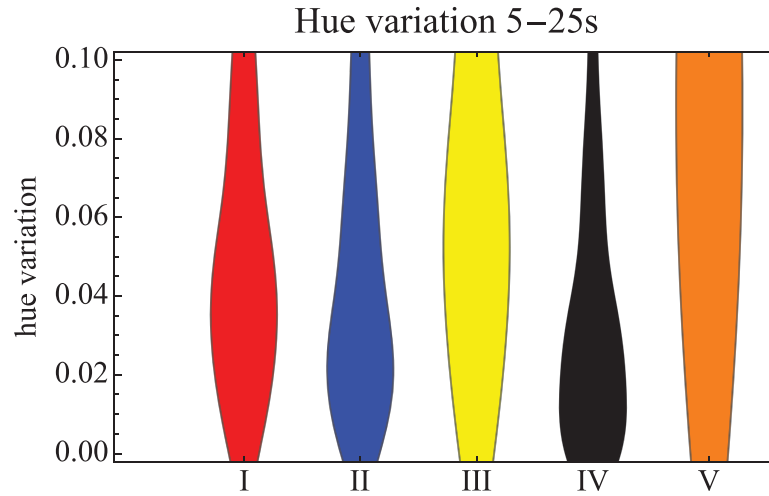


Fig. 12. The distribution of the “wiggleness” in the hue time courses. The hue index is measured on a (periodic) scale of 0–6, thus a variation of 1 is the difference between, for instance, yellow and red or green. In this representation the distributions for interfaces III and V are largely off scale. The differences between interfaces are indeed huge.

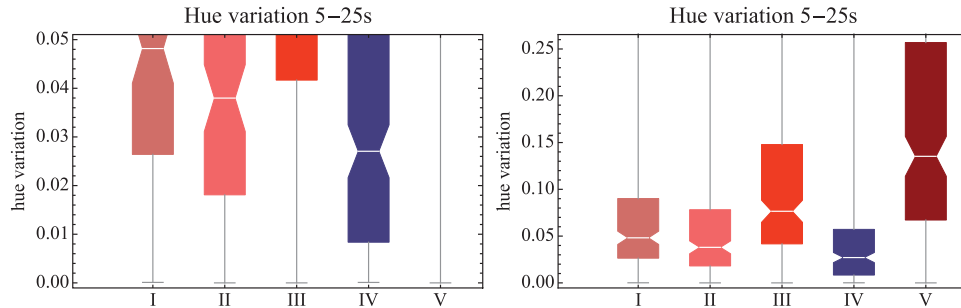


Fig. 13. Quartiles of the wiggleness in the hue time courses. The two graphs differ only in scale. Notice the huge difference between interfaces I, II, and IV as compared with III and V. In the right-hand graph one may judge the distinction between interfaces I, II, and IV; it is of a quantitative nature, with interface IV being lowest. In the left-hand graph one sees the distinction between interfaces I, II, and IV on the one hand, and III and V on the other. This difference is more of a qualitative nature. Especially interface V is seen to lead to very contorted orbits. The median standard deviation of the best interface is about 1/40 of the difference between yellow and red or green. Thus, users may resolve about 100 hues along the hue circle.

Whereas interfaces I and II are roughly on a par (interface I being worse than II because of the aforementioned topological break), interface IV does much better, in that the approach of the GOAL hue is gradual and monotonic.

An overall numerical assessment uses the median and interquartile range (Figure 13). Again the difference between the interfaces is immediately apparent, interface IV being the best by a considerable margin. This is the interface that was taught from kindergarten (using crayons) to universities (using mathematical accounts of physical optics and colorimetry), and used in industry in continental Europe from the 1920s until the 1950s (of course, as a paper atlas instead of an electronic display). It was only abandoned in the post WW II period when Europe adopted many US standards so as to get in sync with the modern world.

A completely different way to represent the results is to consider the orbits in RGB space (Figure 14). It is immediately obvious that the current *RESULT* approaches the *GOAL* in a very roundabout manner. The nature of the orbits is dictated by the nature of the available interactions. This is visually evident from the “city block” structure of the orbits for interface *v*. For interfaces *i* and *ii* the orbits are largely confined to constant hue planes, clearly revealing the nature of the GUI interactions. For interface *iii* the structure is dominated by the opponent directions, and the nonlinear constraints. Notice that the user frequently runs into these constraints, rendering the interface unpleasant to use, because of an “unnatural feel.” For interface *iv* the orbits immediately zoom in to the right hue, then the right neighborhood. The final corrections are largely confined to the *CWK* triangle.

It is hard to quantify the differences through a single number. We find that various measures aimed at capturing the “wiggleness” of the orbits yield very similar results, with color pickers *i*, *ii*, and *iv* having a very significant edge over color pickers *iii* and *v*. Differentiations within these groups tend to depend critically on the exact nature of the measure.

It is of some interest to consider the precision of the final results. In Figure 15 we show data for interfaces *iv* and *v* (approximately the best and the worst). These data are compromised by the fact that our data, as was only to be expected, is full of what are probably best considered “outliers.” Thus, the mean and covariance matrix are perhaps not the most appropriate statistics. Granted this, we still see that much of the variance is in the planes of constant hue.

## 5. CONCLUSIONS

In this study, we compare the initial interactions of 11 naïve users with five rather different color picker GUIs. We are well aware that these 11 users in no way comprise a representative population sample. However, such a sample is a utopic ideal anyway, since there are essentially no people who know how to interact with mouse and keyboard that have not a bias—through experience—for one or more color pickers, whereas such a population sample should probably not include people oblivious of the use of mouses or keyboards. It is a catch-22 situation [Heller 1961]. Eleven naïve, first-time observers are enough of a gamut to allow for useful comparisons to be made.

Five color pickers out of a sheer infinity of possibilities may also seem arbitrary. True enough. Yet the choice is sufficient to allow for interesting comparisons. The present data already serve to illustrate some important aspects of color picker design; the most generic types are represented.

Our mode of presentation differs from that conventionally used in colorimetric studies in that we made an attempt to set up conditions similar to typical use in various applications. As a consequence, many well established ball park measures for just noticeable differences, and so forth, are not applicable. This is clearly shown by our results.

The earliest extensive comparable study is by Schwarz et al. [1987], and the most important follow-up study is by Douglas and Kirkpatrick [1999]. The Schwarz et al. study compares RGB, YIQ, LAB, HSV, and an opponent color model, whereas the Douglas and Kirkpatrick study compares RGB with HSV. The Schwarz et al. study concentrates upon the formal properties of the color systems, whereas the Douglas and Kirkpatrick concentrates on what these authors call “visual feedback.” These studies are very differently oriented. Moreover, their results appear mutually conflicting on first sight. Douglas and Kirkpatrick spend considerable discussion on the explanation of such differences. Perhaps remarkably, our results seem at first blush in blatant conflict with both studies. However, all these apparent conflicts are almost certainly due to methodological and interpretative factors.

One difference concerns supposed intrinsic differences between various parameterizations. The Schwarz et al. study concentrates on this. A problem is that the various “systems” regarded by Smith are even ontologically distinct. Some are colorimetric, others are based on eye measure, and



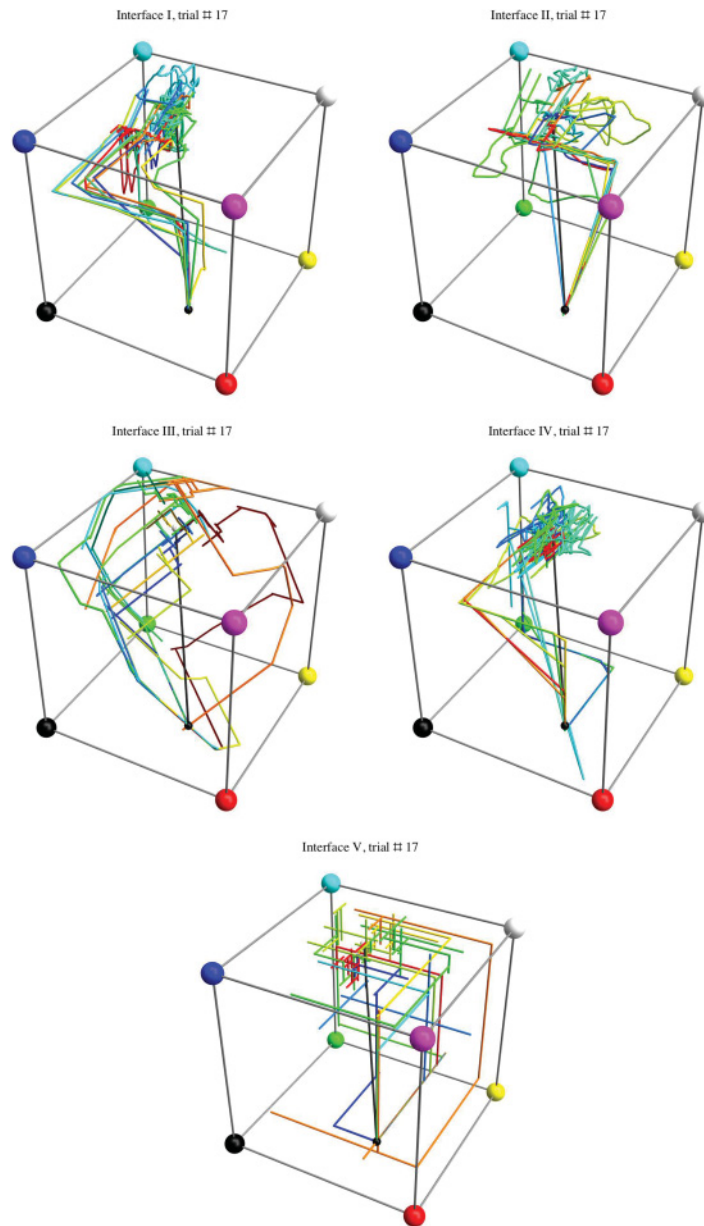


Fig. 14. The orbits from initial RESULT to the GOAL for trial #17, and all interfaces. For each interface the orbits for all observers are presented, each observer in a distinct hue. The initial (forced) RESULT is indicated by the small black sphere, and the GOAL by a gray sphere (occluded by the tails of the orbits). The shortest route in RGB space in the Euclidean metric is indicated by the black line. (Notice that the vertices are colored. A red color indicates the red  $(1, 0, 0)$  vertex of the RGB cube and so forth.)

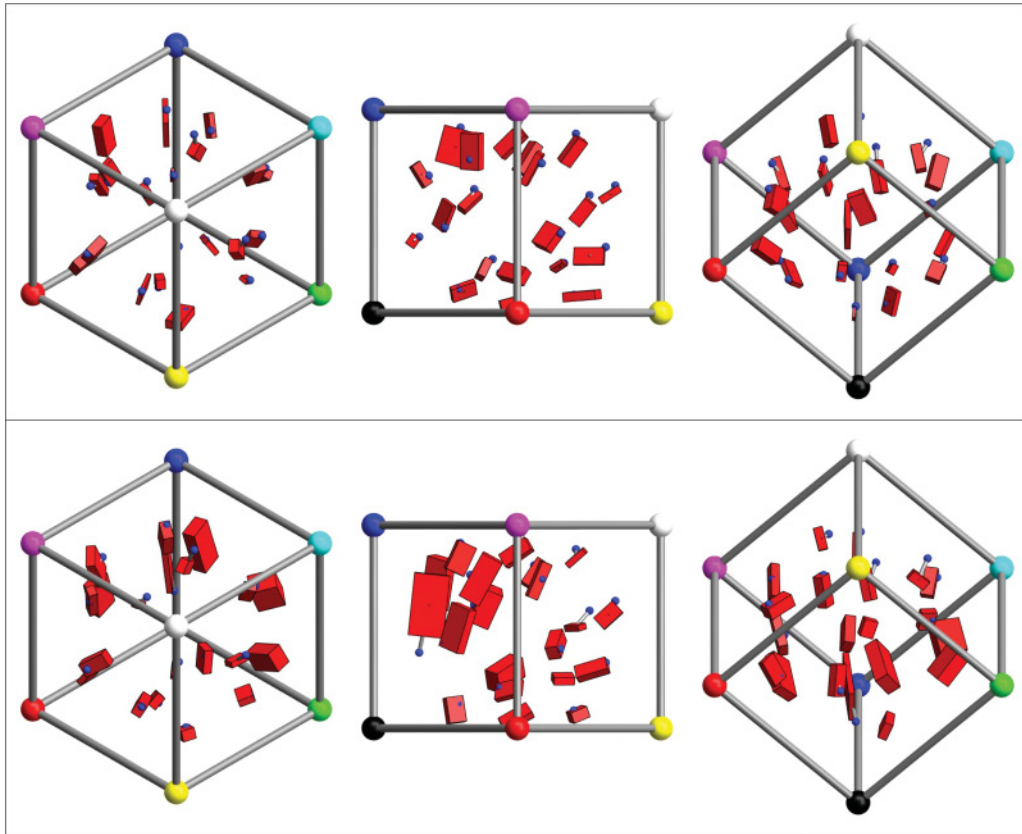


Fig. 15. The means and covariances for interfaces IV (top row) and V (bottom row). The “RESULT” has been defined here as the list of achieved colors during the final 20% of the full period spent by the user, pooled over all observers. The red blocks are centered at the mean, their attitude and shape defined by the eigensystem of the covariance matrix. We show views of the RGB cube as seen from the  $\{1, 1, 1\}/\sqrt{3}$  (left column),  $\{1, -1, 0\}/\sqrt{2}$  (center column), and  $\{1, 1, -2\}/\sqrt{6}$  (right column) directions. (Notice that these directions are mutually orthonormal.) The GOALS are indicated by the blue points. Thus, the final RESULTS are scattered about a location that is shifted with respect to the GOAL. The data are rather noisy though. (Notice that the vertices are colored. A red color indicates the red  $(1, 0, 0)$  vertex of the RGB cube and so forth.)

still others are specific to RGB monitors or video encoding. In a comparison of the previous results with ours, one should disregard all but the RGB, HSV, and perhaps the opponent systems.

Another difference is with regard to the very notion of “user interface.” In the Schwarz et al. study, RGB and HSV are treated as intrinsically distinct, in a formal sense, quite apart from the actual implementation. Thus, “RGB is better than HSV” (a significant result reported by Schwarz) is considered a meaningful statement even in the absence of the description of the GUIs. Douglas and Kirkpatrick do consider this. Their “visual feedback” pertains to the GUI. For instance, a RGB interface with three uniformly gray sliders is a GUI without visual feedback, whereas a GUI with sliders that are colored with gradients from black to red, green, and blue offer feedback. It is hardly a surprise that “offering visual feedback” makes a major difference. In our study, all GUIs offer full visual feedback, so the issue does not arise.

An interpretative factor has to do with the notion of “difference.” Schwarz et al. report “significant” differences. A closer inspection reveals that these differences are indeed statistically significant, but

really very minor. As compared to the considerable differences found by us, it would be more appropriate to say that Schwarz et al. did not detect any differences at all. Douglas and Kirkpatrick do not find significant differences between RGB and HSV; the only—and indeed major—differences they find are between visual feedback and no visual feedback.

The most important differences between the earlier studies and ours are methodological. They are of three types.

First of all, the earlier studies implemented a split-field *matching task*, the result and goal being abutting patches. The observer's task is then to make these patches equal, which implies that the edge between the two uniform areas should vanish. This does not involve color as a quality, nor short term color memory. This is very different from our paradigm, in which the user needs to keep the GOAL in mind, perhaps, occasionally “refreshing” the memory. In our case, color as a *quality* matters. We consider this very important in interface studies because it most closely reflects normal use, the matching task for abutting patches being rather artificial.

Secondly, perhaps most importantly, we studied only naïve, first-time users. In both prior studies users were trained prior to trials. They received instructions, and were allowed a training period. This makes a huge difference. Both prior studies report significant learning effects. A trained, expert user will work fine with any interface; in order to show up interface differences, one really needs to study naïve, first-time users.

Finally, it makes a difference which colors are considered for the GOAL, and from which color the hunt for the goal is started. This is well understood by Douglas and Kirkpatrick, who speculate that the result of Schwarz et al. might be due to the fact that the GOAL tended to be close to a red, green, or blue. These authors themselves decided on a set of conventional colors, often used in photographic applications, which they consider as typical for environmental color gamuts. In our study, we made an attempt at a uniform sampling of RGB space, although the choice is necessarily limited.

As a consequence of these differences, it is not really useful to consider the apparent differences in the findings. The major difference in our study, as compared to the prior ones, is that we find a much shorter effective period (at most half a minute to completion), and large—not just “statistically significant”—differences between GUIs.

It is of some interest to notice that the differences that show up in our study closely fit the prior expectations of the earlier authors. One expects interfaces I, II, III, and IV to be much “better” than interface V, because that is why they were designed the way they are to begin with. Yet they paradoxically showed that the RGB sliders interface beats all others. The authors go to some length to explain why their studies fail to reflect common wisdom.

It is perhaps encouraging that there are also many common findings in the work of Schwarz et al., Douglas and Kirkpatrick's, and ours. Here are the major agreements, some more implicitly, others explicitly mentioned by the earlier authors:

- Learning is important. An expert user will be happy with any tool, as long as it is a familiar one. Novel users soon get better with continued use of a GUI.
- Approaching the target proceeds by stages. The first stage is a jump to the general neighborhood of the target. The second stage (the crucial 20s) is used to navigate to the perceptual neighborhood of the target. Finally, the third stage is an erratic hunting behavior that never does converge. It merely leads to frustration, since the user is still conscious of being “off.”
- Users hate, and are seriously impeded by, hidden constraints—most typically dead zones. They fail to foresee these, and are frustrated in actions that “should work.” They experience the interface as somehow “broken.”

—Good GUI design tends to make a lot of difference to the first-time user. Expert users deal routinely with even very haphazardly thrown together interfaces.

The typical manner of approach of the current RESULT to the GOAL, such as that illustrated in Figure 10, is also apparent in the work of both Schwarz et al., and of Douglas and Kirkpatrick. Especially an initial fast approach, the existence of apparent plateaus in the zooming-in phase, and a final erratic behavior are common in all studies. It is not necessarily the case that this reflects phases in user strategies, though. For instance, very simple models that are merely improvements of random search also exhibit these phases.

We implemented “harmony finding” [Geem et al. 2001], a generic “soft computing” algorithm. One initially collects a number of random samples, and orders them with respect to mismatch. At each iteration one replaces the worst 50% with novel samples, and slightly perturbs the best ones, always retaining a fixed number of overall best matches. Such an algorithm searches the whole space, thus cannot be stuck in a suboptimum, and explores the immediate neighborhood of the good matches as a kind of random hill climbing. The algorithm is stopped when either a minimum criterion, or a maximum number of iterations is reached. This method immediately reproduces the initial phase, and the zooming-in phase with plateaus. It yields a broad distribution of times to criterion. Such a method does not capture the final, erratic phase, for the distance to GOAL is monotonically decreasing by design. This can easily be implemented by changing the estimation of the match, or by limiting the memory span of the method.

Thus, the nature of the approach of the GOAL is already captured by the simplest of mechanisms. It is likely that the human user manages to profit from the GUI design though. This is evident from the fact that the soft computing algorithm cannot do better than with the RGB interface, which samples the RGB cube in a uniform manner, whereas this GUI is one of the least efficient for the naïve, first-time human user. Indeed, we have shown that the approach to the GOAL immediately reflects specific properties of the GUIs.

Most of the findings we report are rather obvious from generic UI (not just color picker oriented) design principles. Thus, we find that a mere triplet of linear RGB sliders, that is, perhaps the “generic color picker,” is one of the worst possible GUIs for the naïve user, despite the fact that it might appear “natural” and is certainly a tool one would not do without, for the professional. The reason is obvious enough. Naïve users lack a mental model of RGB space, and thus cannot foresee manifold interactions between the GUI elements. For example, a unique quality like “yellow” is coupled to a pair (red and green), instead of a single slider. The result is usually an aimless hunting behavior.

From a formal perspective, a slightly different interface, although perhaps appearing similarly unpleasant to the naïve user, is offered by the Hering-type GUI [Hering 1964]. This is especially interesting, because the Hering “opponent” representation is generally considered “most natural” in phenomenologically oriented circles [Hurvich and Jameson 1957]. Whereas we do not necessarily disagree with such notions, we have not been able to implement a Hering-type interface that does not involve very complicated interdependent constraints, causing unexpected “dead zones” in slider ranges. Our implementation may well be one that is most kind to the naïve user, yet all users hate it. This is also a generic finding: interaction domains should be free of dead zones, and interaction domains should be mutually orthogonal.

A very clear-cut finding is that any two-plus-one-dimensional interface beats any triple one-dimensional interface hands down. The two dimensions are explored through arbitrary mouse movements. The implementations of the two-plus-one-dimensional interface in our color pickers I, II, and IV are very different, yet they are roughly equally efficient. Looking in detail reveals obvious differences though. Especially color picker I leads to frequent unnecessary spikes due to a bad graphical

design: the interface does not reflect the topology of the relevant dimension. A similar problem occurs as the user passes near the center of the disk in color picker II.

Another point, related to the two-dimensional GUI elements, is the presence, or absence, of anchoring features. Examples of the problems induced by insufficient anchoring are seen in color picker II, where users do not take the exact location of the center into account. As a result, the hue may suddenly jump to its complement. The problem here is that users treat the interaction areas as uniform, whereas it is really far from that: the center and its environment are singular because the hue becomes indeterminate.

For color picker IV, based on the structure of the Ostwald [1917, 1919] color atlas, we find a smooth, gradual approach of the GOAL, whereas this approach is rather more fluctuating for the other interfaces. This may well explain the informal fact that all observers “liked” interface IV best, whatever that may be construed to mean.

One issue not considered in this study involves *learning*. Our naïve observers completed the tasks only once. We did not plan on repeated sessions because any session may also be considered to be an unsupervised training session. No doubt a user will get better with any GUI given sufficient experience. Explicit instructions will have dramatic effects, for example, explaining some basic properties of RGB colors will soon enable users to use the RGB-slider interface. Even mere experience will change the user. One aspect of the merits of GUIs thus involves the ease by which naïve users learn to use them. There is little doubt that the color pickers considered in this article mutually differ greatly in this respect. Possibly initial user preference is an indication. We have left this aspect unexplored, as it would require a major investment in user time. It is no doubt of considerable importance though, and should be included in implementation decisions. For applications that are used only infrequently—also quite common—the issue hardly arises, of course.

In conclusion, we find that the best color pickers aimed at naïve or first-time users are of the two-plus-one, rather than triple one-dimensional variety. Users approach the target in the smoothest manner with the Ostwald-type, or HWB interface. This agrees with their informal appreciations, and it might well influence learning behavior. In terms of efficaciousness color pickers I, II, and IV are on a par. Interface IV yields the smoothest, monotonic approach of the GOAL hue. Apparently, Smith and Lyons [1996] intuitions were right on the dot.

As a bottom line, given a good GUI naïve users pick a color to an accuracy of roughly 0.05–0.1 in R, G, and B coordinates in about 20–30s. The initial 20s are decisive; additional efforts are hardly effective in obtaining a more precise result.

## A. THE PARAMETERIZATION OF THE COLOR PICKERS

The various color systems proposed for screen colors are, of course, grounded in the colorimetric systems designed for generic contexts [Newton 1998; Maxwell 1860; Ostwald 1917, 1919; Kelly and Judd 1976; Wyszecki and Stiles 1928]. They also purport to have a relation to phenomenology [von Goethe 1982; Runge 1810; Munsell 1905, 1912; Ostwald 1917, 1919]. A relation between these various, ontologically disparate topics was suggested by Schopenhauer [1994]. For a quick introduction, see Poynton [1997] or MacEvoy [2014].

The screen colors (here “color” denotes a class of spectral compositions) are generated via the so-called *primary colors*, which are

- red*: an emission primarily at the long wavelength end of the visual spectrum;
- green*: an emission primarily at the mid of the visual spectrum; and
- blue*: an emission primarily at the short wavelength end of the visual spectrum.



Thus, the screen colors are parameterized by their red (R), green (G), and blue (B) coordinates [Foley et al. 2005]. Thus, we will specify a RGB color as  $\{r, g, b\}$ . These coordinates run from 0 to 1 (say), where zero is a natural origin (no light), whereas 1 is due to technical constraints. (Many systems will use the range 0–255, but it makes no difference to the present discussion.) With appropriate choices of the spectra of the three primary colors, they will appear “red,” “green,” and “blue” to the generic human observer. These appearances are visual qualities that are mental attributes. These are Schopenhauer’s “parts of daylight” [Schopenhauer 1994]. Moreover, the sum of the three primary colors (parts of white) will appear “white” (W), whereas the *secondary colors* will appear as

*yellow*:  $Y = R \cup G$ ; “yellow” is neither “reddish,” nor “greenish”;  $RGB = \{1, 1, 0\}$ ;  
*cyan*:  $C = G \cup B$ ; “cyan” is neither “greenish,” nor “bluish”;  $RGB = \{0, 1, 1\}$ ; and  
*magenta*:  $M = B \cup R$ ; “magenta” is neither “bluish,” nor “reddish”;  $RGB = \{1, 0, 1\}$ .

A lot might be said about this (e.g., the choice of primary colors also determines the available color gamut), but here we simply accept the choice made by the manufacturer. In practice, display units are seen to converge on very similar characteristics. This is largely due to a variety of physical and material boundary conditions. Indeed, it is unlikely that the adoption of a different brand would affect our findings to any significant extent.

The periodic sequence YGCBMR derives its topology from the fact that the hue of equal mixtures of left and right neighbors in the sequence matches that of the fiducial item (Ostwalds “Principle of Internal Symmetry” [Ostwald 1917, 1919]), and that items half a period apart add to white, where white is the addition of the three primary colors. More precisely, one has  $K = 0R + 0G + 0B$ ,  $R = 1R + 0G + 0B$ ,  $G = 0R + 1G + 0B$ ,  $B = 0R + 0G + 1B$ ,  $C = 0R + 1G + 1B$ ,  $M = 1R + 0G + 1B$ ,  $Y = 1R + 1G + 0B$ , and  $W = 1R + 1G + 1B$ . The periodic sequence is often referred to as the “color circle,” though it is more properly understood as a *hexagon*. A hexagon has the topology of the circle, but straight edges, that is to say, a stretch like Y–G is linear, not curved. This becomes evident from the following analysis. The preceding combinations, originally identified by Schopenhauer in the early 19th century, essentially exhaust the categorically distinct qualities. The gamut of all colors is contained in their convex hull.

Consider a specific color; we use  $x = \{0.7, 0.3, 0.1\}$  as an arbitrary example (Figure 16). What does it look like? In order to find out, you may start by “subtracting the white content.” Notice that  $w = \{1, 1, 1\}$ , so you can only subtract  $0.1w$ , for the RGB coordinates are nonnegative. Notice that  $w = \min[x]$ . You are left with  $x' = \{0.6, 0.2, 0\} = x - wW$ , where  $w$  (white content) equals  $w = 0.1$ . At least  $x'$  is not diluted by white, but unfortunately  $x'$  is not as bright as could be.

In order to handle the brightness issue you must also consider a “black content.” Analogous to the definition of the white content you set  $k = 1 - \max[x]$ . Here “ $k$ ” comes from “key”: in the printing business the black plate was referred to as “key.” (Using the letter “ $k$ ” conveniently distinguishes it from the letter “ $b$ ,” already reserved for blue.) Apparently,  $k = 0.3$  in our example. One way to “remove the black” is to scale  $x'$  by the maximum amount, yielding  $x'' = \{1, 1/3, 0\}$ . This is a “pure color” without any admixture of white and black. It is technically known as a “full color.” A full color is the brightest, and most saturated color of a given hue. Full colors and hues stand in a 1:1 relation.

Thinking of a color as the sum of white, black, and full color, you may define the “color content”  $c$  through the constraint  $c + w + k = 1$ . Thus,  $c = \max[x] - \min[x]$ . It is the amplitude of the RGB variation. In the example,  $c = 0.6$ .

With the introduction of the color, white, and black contents, you can express our example color as  $x = cF + wW + kK$  (where  $K = \{0, 0, 0\}$ ). The “full color”  $F$  is seen to be  $F = \{1, 1/3, 0\}$  in the example. The full color lies on the color circle, so you can uniquely specify it by its location on the circle  $h$  (the “hue index”), whereas the color  $x$  is obtained from  $F$  through the addition of some white and some black.

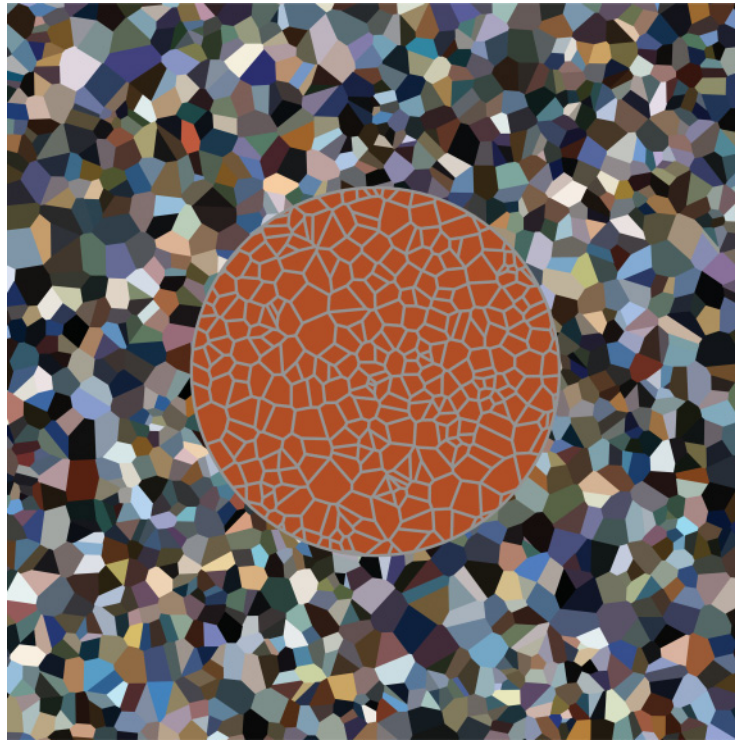


Fig. 16. An RGB color as presented in the experiment (either GOAL or RESULT). This example is the color  $x = \{0.7, 0.3, 0.1\}$  discussed in the Appendix. It is presented as a uniform disk, on which a random cobweb pattern of thin medium gray lines is superimposed (for a good impression this needs to be presented on an electronic display, at the right size, and so forth—in the printed figure the lines have been thickened for clarity). The pattern gives the patch a more “material,” “opaque” look; uniform patches tend to look “ethereal,” they have no proper surface. The patch is presented on a randomly textured background, which acts like a context that anchors the RGB scale. The colors of the background have been generated such as to represent a “natural color gamut,” roughly mimicking the statistics of some data bases of natural colors.

The technical implementation is left open: on a CRT monitor you get black “for free,” obtaining white by emission of radiation, whereas on a printer you get white “for free,” obtaining black by depositing ink [MacDonald and Loweki 1997; Yule and Fields 2001]. Thus, on a monitor the equation  $x = cF + wW + kK$  simplifies to  $x = cF + wW$ , whereas on a printer it simplifies to  $x = cF + kK$ . On a monitor  $W$  might be achieved by superposition of  $R$ ,  $G$ , and  $B$ , whereas on a printer  $K$  might be achieved by superposition of  $C$ ,  $M$ , and  $Y$ . This is by no means necessary though: a monitor might have an additional white emitter, and a printer an additional black ink. The latter addition is common; the former occasionally reemerges as an innovation.

The full color  $F = \{1, 1/3, 0\}$  has one coordinate 1, one coordinate 0, and a third coordinate in the range  $[0,1]$ . You may check that this is true for any choice of  $x$ . This implies that any full color is a linear combination of a primary and an adjacent secondary color. In the example,  $x'' = (1 - a)R + aY$  (i.e.,  $(1 - a)\{1, 0, 0\}$  plus  $a\{1, 1, 0\}$ ), where  $a = 1/3$ . Thus,  $x''$  lies on the edge  $RY$  or the YGCBMR hexagon, one-third of the edge length from  $R$ . Thus,  $F$  is a *yellowish red*. When you index YGCBMR with 012345 (periodically, thus “6” again becomes 0),  $F$  gains the “hue index” 5.33.... Then the color  $x$  becomes hue 5.33..., with 0.6 color content, 0.1 white content, and 0.3 black content, which may be rendered in words as a “*yellowish shade of red*” (Figure 16). “Shades” are mixtures of full color with black,

whereas “tints” are mixtures with white. Informally, one calls colors with much black “shades,” and those with much white “tints,” although the former usually have white content too, and the latter some black content. On the RY edge of the color circle you may verbally differentiate “red,” “yellowish-red,” “red-yellow,” “reddish-yellow,” and “yellow,” and analogously for the other edges. Some hues have conventional names, thus “red-yellow” is usually known as “orange” [Kelly and Judd 1976].

The HCWK denotation ( $h$ : hue index;  $\{c, w, k\}$  color, white, and black content) is the Ostwald specification of a color. Because of the constraint  $c + w + k = 1$ , the HCWK denotation specifies three degrees of freedom. Of course, HCWK and RGB are fully equivalent. However, the “straightforward” implementation leads to color pickers IV and V, with a decidedly different “feel,” and very different efficaciousness for naïve observers. One feels that HCWK is closer in specifying the “natural mental representation” of naïve users than RGB is. The HCWK notation (called “HWB”) was reinvented in 1996 by Alvy Ray Smith, of computer graphics fame [Smith and Lyons 1996]. Smith developed the HSV and HSL CG color models [Smith 1978], in general use today, but in the abstract of the 1996 paper one reads:

The two most common color selector models, other than RGB . . . , are the hue-based HSV . . . and HSL . . . color models. It is shown that both of these models are flawed. A closely related model, HWB (Hue-Whiteness-Blackness), is introduced that avoids the flaws, . . . , and is very easy to teach to new users: Choose a hue. Lighten it with white. Darken it with black.

This is nothing but the Ostwald system (technical details in Koenderink [2010a]) for RGB, albeit almost a century after Ostwald introduced it. The Ostwald system, despite its prominent position in central European color science, is not even mentioned in the Smith and Lyons paper—these authors truly reinvented the wheel. (Of course, by no means a minor achievement!) Perhaps unfortunately, the HWB color space never muscled out HSL, or HSV, as its authors intended.

Similarly, color pickers I, II, and III are based on equivalent, but distinct parameterizations. The parameterizations may be based on phenomenological considerations (as in the Hering picture leading to color picker III), or considerations that are based on colorimetric quantities relating to irradiating spectral power density, or the radiance of surface scattered beams relative to the radiance of the radiant source.

*Color Picker I.* Color picker I is conceptually based on a particular “hue-saturation-value” type of parameterization. Various definitions of these parameters may be found in the literature. Naïve users have no notion of their formal definitions, which is why we did not follow any “standard.” The formalization used is somewhat intuitive, once the structure of RGB space is understood. Various changes would probably change little or nothing to our findings.

The hue is simply parameterized by the hue index discussed previously. The saturation ( $s$ ), and value ( $v$ ) are defined as  $v = \max[r, g, b]$ , and  $s = 1 - \min[r, g, b] / \max[r, g, b]$ . Notice that this does not involve any constraints on  $r$ ,  $g$ , and  $b$  apart from positivity. This parameterization is best suited for radiance, rather than scattered radiance relative to incident irradiance [von Helmholtz 1867; MacEvoy 2014]. The saturation-value description does not involve “black” (or  $k$ ) at all. Only the additional relation that  $r$ ,  $g$ , and  $b$  should be on  $[0, 1]$  forces values to be limited.

Introducing  $c + w + k = 1$ , one readily derives  $s = c / (c + w)$ , and  $v = c + w = 1 - k$ . Thus, saturation is like a “modulation depth,” and value the “nonblack fraction.” These relations are nonlinear, thus color pickers I and IV (see the following) are guaranteed to have a very different “feel” to them.

*Color Picker II.* Color picker II is not much different from color picker I. Conceptually, it uses “lightness”  $p$  instead of value, and “saturation”  $q$ , defined by way of the relation

$$x = p [q^F + (1 - q)W],$$

which should be fairly intuitive. Saturation interpolates linearly between the full color and white, whereas lightness measures overall amplitude. This expression shows that saturation is limited to  $[0,1]$ , whereas lightness is only required to be nonnegative: this is again essentially a spectral radiance description.

Introducing the constraint  $c + w + k = 1$ , one easily derives that  $p = c + w = 1 - k$  and  $q = c/(c + w)$ . These turn out to be the same relations as in the previous case.

In this article, we prefer HWB throughout; we do not bother with either HSV or HSL proper. The only real difference between color pickers I and II is in the *graphics*. This enables us to compare color pickers I and II rather directly.

There is a variety of different parameterizations much like those used in color pickers I and II, that are in common use [Smith 1978; Joblove and Greenberg 1978; Foley et al. 2005; Levkowitz and Herman 1993; Agoston 2005]. For instance, lightness is frequently defined as  $(r + g + b)/3$  instead of  $\max[r, g, b]$ , and so forth. This is extensively discussed in the CG literature. We have not included such additional instances in this study, because naïve users will hardly notice the difference.

*Color Picker III.* Color picker III is based on Hering’s notion of “opponent colors” [Hering 1964]. The idea is that the “natural dimensions” for the human observers are black-white, red-green, and yellow-blue balances. This may be formalized in a number of ways, each with distinct pros and cons. Hering never specified a formalism. A number of different formalisms may be set up to capture Hering’s original intentions [Bratkova et al. 2009]. We used one of the simplest possible:

$$r = kw + \frac{1}{2}rg + \frac{1}{3}yb, \quad g = kw - \frac{1}{2}rg + \frac{1}{3}yb, \quad b = kw - \frac{2}{3}yb,$$

which is equivalent to

$$kw = \frac{1}{3}w, \quad rg = r - g, \quad yb = \frac{1}{2}y - b.$$

We augment these relations with the constraints that  $r, g, b$  are on  $[0,1]$ . Of course, Hering never considered such constraints; the “opponent” description equally applies to spectral radiance. Although the preceding relations look simple enough, because linear, there are actually severe problems because of the range constraints on the parameters. Notice that  $r, g, b$ , and  $kw$  are on  $[0,1]$ , whereas  $rg$ , and  $yb$  are on  $[-1,+1]$ . But let  $rg = 1$ , then  $yb$  is restricted to  $[-1/2,+1/2]$ , and  $kw$  to  $[0,2/3]$ . It is such constraints that spoil the natural “feel” of this interface. The modal dead zones confuse the user.

*Color Picker IV.* Color picker IV is a simple GUI based on HCWK. This parameterization was explained previously. Because of the basic relation  $c + w + k = 1$ , this description is specifically tailored to object colors [Koenderink 2010a]; it does not apply to spectral radiance, unless one introduces a reference “illuminant.” This GUI especially appeals to visual artists, used to “lighten” a (full) color with white, and “darken” it with black.

*Color Picker V.* Color picker V is perhaps the simplest GUI, with three sliders that separately control the red, green, and blue amounts. One imposes the constraints that  $r, g$ , and  $b$  are on  $[0,1]$ , usually regarded as a mere technical limitation. This GUI is okay for—perhaps even preferred by—professionals, whereas naïve users meet with the problems mentioned in the main text.



## B. THE GRAPHICAL IMPLEMENTATION OF THE COLOR PICKERS

See Figure 3.

*Color Picker I.* The GUI consists of a rectangular interaction area, and a slider. The slider controls the value; the rectangle is the Cartesian product of the hue scale and the saturation. Both the value and the saturation are in  $[0,1]$ , thus map naturally on a linear slider, and a side of a rectangular area. The hue scale is problematic because it is periodic. This forces one to mentally identify two opposite sides of the rectangle. Of course, the hue index and the saturation are incommensurable, thus the horizontal and vertical dimensions of the rectangle cannot be compared. We map it graphically on a square, because this optimizes the resolution of mouse movements in the plane. The square is colored with the corresponding color for unit value. It needs getting used to; there is little that is “natural” about the presentation. Providing anchors would no doubt be useful. However, we implemented the conventional smooth graphics.

*Color Picker II.* The GUI consists of a circular disk and a linear slider. The slider controls the lightness, and the disk is a polar map of the Cartesian product of the color circle and the saturation scale (a linear segment). The origin of the saturation scale coincides with the center of the disk, thus the radius parameterizes saturation, the angle hue. The perimeter of the disk has the full colors; toward the center all colors pale to white. This visually identifies the center, though only in a very coarse way. Providing some anchors would certainly be welcome. However, we implemented the conventional smooth graphics.

*Color Picker III.* The GUI consists of just three linear sliders. Two of these are bipolar ( $by$  and  $rg$  vary on  $[-1, +1]$ ), and the third one is unipolar ( $kw$  on  $[0,1]$ ). Since the sliders can be color coded, their functions are immediately visually obvious. Most of the problems are due to the interdependent constraints that cause “dead regions” in the slider domains.

*Color Picker IV.* The GUI consists of an annulus and an equilateral triangle. The annulus presents the color circle. Since it is colored with the full color range its function is immediately visually obvious. The triangle has  $F$ ,  $w$ , and  $k$  as vertices. Because the interior is colored, its function is again immediately visually obvious. The mouse position in the triangle implements the  $c + w + k = 1$  constraint in a natural manner, for if you place  $cwk$  weights at the corresponding vertices, the center of gravity is the mouse position. These are the barycentric coordinates, originally introduced in geometry by Möbius [1827]. The two interface elements function fully independently of each other. In this example, we provided anchors in both the color circle and the triangle. There are various ways to do so; the present one is probably not optimal, though certainly better than a smooth distribution (as in the implementation of Corel’s Painter application [Zimmer 2012]).

This GUI is not in common use, probably because no platform natively supports annular sliders, or triangular interaction areas.

*Color Picker V.* The GUI consists of just three linear sliders. All are unipolar ( $r$ ,  $g$ , and  $b$  on  $[0,1]$ ). Since the sliders can be color coded, their functions are immediately visually obvious. All domains are uniform; there are no dead zones. This is an easy interface if you have the structure of the RGB cube in mind. However, for naïve observers there exist many unexpected interactions due to the fact that various “unique hues,” like yellow, depend on “simultaneous” slider settings. The topology of the hue domain also poses surprises. That decreasing the red and green sliders simultaneously often yields a similar effect to increasing the blue slider is something that beats most first-time users.

## ACKNOWLEDGMENTS

The first two authors are grateful for Karl Gegenfurtner’s hospitality during our stay at the University of Giessen where we finalized the manuscript.



## REFERENCES

- Max K. Agoston. 2005. *Computer Graphics and Geometric Modeling: Implementation and Algorithms*. Springer, London.
- Margarita Bratkova, Solomon Boulos, and Peter Shirley. 2009. oRGB: A practical opponent color space for computer graphics. *IEEE Computer Graphics and Applications* 29, 1 (2009), 42–55. Special issue title on generating 3D building models a VR playground for teaching math archive.
- Barbara Brundage. 2012. *Photoshop Elements 11: The Missing Manual*. O'Reilly Media, Inc., Sebastopol, CA.
- Brad Cox. 1983. The object oriented pre-compiler: Programming smalltalk 80 methods in C language. *ACM SIGPLAN Notices* 18, 1 (1983).
- 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 Transactions on Graphics* 18, 2 (1999), 196–127.
- James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. 2005. *Computer Graphics, Principles and Practice* (2nd. ed.). Addison-Wesley Publishing Company, New York.
- Zong Woo Geem, Joong Hoon Kim, and G. V. Loganathan. 2001. A new heuristic optimization algorithm: Harmony search. *Simulation* 76, 2 (2001), 60–68.
- Joseph Heller. 1961. *Catch-22*. Simon & Schuster.
- Ewald Hering. 1964. *Outlines of a Theory of the Light Sense*. Harvard University Press, Cambridge, MA.
- Aaron Hillegass. 2008. *Cocoa Programming for Mac OS X* (3rd ed.). Addison-Wesley, Cambridge, MA.
- Leo M. Hurvich and Dorothea Jameson. 1957. An opponent-process theory of color vision. *Psychological Review* 64, 6 (1957), 384–404.
- George H. Joblove and Donald Greenberg. 1978. Color spaces for computer graphics. *Computer Graphics* 12, 3 (1978), 20–25.
- Kenneth L. Kelly and Deane B. Judd. 1976. *Color. Universal Language and Dictionary of Names. Special Publication 440*. National Bureau of Standards (U.S.), Washington.
- Jan J. Koenderink. 2010a. *Color for the Sciences*. MIT Press, Cambridge, MA.
- Jan J. Koenderink. 2010b. The prior statistics of object color. *Journal of the Optical Society of America A* 27, 2 (2010), 206–217.
- Haim Levkowitz and Gabor T. Herman. 1993. GLHS: A generalised lightness, hue and saturation color model. *CVGIP: Graphical Models and Image Processing* 55, 4 (1993), 271–285.
- David L. MacAdam. 1943. Specification of small chromaticity differences. *Journal of the Optical Society of America* 33 (1943), 18–26.
- Lindsay W. MacDonald and Anthony C. Loweki. 1997. *Display Systems*. John Wiley and Sons.
- Bruce MacEvoy. 2014. An Extensive Site about Color Science and Paint Mixing on the Web. Retrieved April 17, 2014 from ColorVision.handprint.com.
- James Clerk Maxwell. 1860. On the theory of compound colours, and the relations of the colours of the spectrum. *Philosophical Transactions of the Royal Society of London* 150 (1860), 57–84.
- Alfred Ferdinand Möbius. 1827. *Der Barycentrische Calcul*. Johann Ambrosius Barth, Leipzig.
- Albert H. Munsell. 1905. *A Color Notation*. G. H. Ellis Co., Boston.
- Albert H. Munsell. 1912. A pigment color system and notation. *Journal of Psychology* 23, 2 (1912), 236–244.
- Isaac Newton. 1998. *Opticks or, a Treatise of the Reflexions, Refractions, Inflexions and Colours of Light: Also Two Treatises of the Species and Magnitude of Curvilinear Figures. Commentary by Nicholas Humez*. Octavo, Palo Alto, CA. Opticks was originally published in 1704.
- Wilhelm Ostwald. 1917. *Die Farbenfibel*. Unesma, Leipzig.
- Wilhelm Ostwald. 1919. *Der Farbkoerper und Seine Anwendung zur Herstellung Farbiger Harmonien*. Unesma, Leipzig.
- Charles Poynton. 1997. Retrieved April 17, 2014 from FrequentlyAskedQuestionsAboutColor. A self-published Frequently Asked Questions page on the Internet. <http://www.poynton.com/ColorFAQ.html>.
- Charles Poynton. 1998. The rehabilitation of gamma. In *Proceedings of SPIE (3299)*. SPIE, Washington, 232–249.
- Phillip O. Runge. 1810. *Die Farben-Kugel, Oder Construction des Verhaeltnisses Aller Farben Zueinander*. Perthes, Hamburg.
- Arthur Schopenhauer. 1994. *On Vision and Colors*. Berg, Providence.
- M. W. Schwarz, W. B. Cowan, and J. C. Beatty. 1987. An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models. *ACM Transactions on Graphics* 6, 2 (1987), 123–158.
- Alvy Ray Smith. 1978. Color gamut transform pairs. *Computer Graphics A* 12, 3 (1978), 12–19.
- ACM Transactions on Applied Perception, Vol. 13, No. 3, Article 13, Publication date: April 2016.

- Alvy Ray Smith and Eric Ray Lyons. 1996. HWBA more intuitive hue-based color model. *Journal of Graphics, GPU, and Game Tools* 1, 1 (1996), 3–17.
- Johann W. von Goethe. 1982. *Theory of Colours*. MIT Press, Providence. Translation by Charles Lock Eastlake.
- Hermann von Helmholtz. 1867. *Handbuch der Physiologischen Optik*. Leopold Voss, Leipzig.
- Günther Wyszecki and Walter S. Stiles. 1928. *Color Science* (2nd ed.). John Wiley & Sons.
- John A. C. Yule and Gary G. Fields. 2001. *Principles of Color Reproduction: Applied to Photomechanical Reproduction, Color Photography, and the Ink, Paper, and Other Related Industries*. GATF Press.
- Mark Zimmer. 2012. Relativistic observer: Color. Retrieved April 23, 2014 from <http://relativisticobserver.blogspot.nl/2012/03/color.html>.

Received October 2015; revised January 2016; accepted January 2016