June 2022

Identifying a Color with Sufficient Contrast Against Two Distinct Background Colors

Peter Kasting

Peter Boström

Identifying a Color with Sufficient Contrast Against Two Distinct Background Colors

ABSTRACT

A good user interface design provides user interface elements that contrast sufficiently with their surroundings. While color contrasting is relatively simple to do when there is one surrounding color, when choosing foreground colors between two background colors, the foreground must simultaneously contrast with both background colors. Given two background colors and an initial foreground color, this disclosure describes techniques to automatically select a foreground color that contrasts with both background colors and is similar in hue to the initial foreground color. The choice of colors can be constrained to a palette of valid colors grouped by similar hue. Given a range of potential relative luminances and two points within that range, the most contrasting luminance to both points is determined to be an endpoint of the range or the midpoint between the two luminances.

KEYWORDS

- Color contrast
- Background color
- Foreground color
- Focus ring
- Color palette
- Accessibility
- Standard RGB space
- HSL (hue, saturation, lightness) color
- Web content accessibility guidelines (WCAG)
- Accessibility
- Relative luminance
- Contrast ratio

BACKGROUND

The color of a user interface (UI) element is optimally such that it contrasts sufficiently with its surroundings. An example is a state indicator which, to be visible to the user, is shown in a color that contrasts sufficiently with the background. Color contrasting is relatively simple to do when there is one surrounding color, but for state indicators at the border between colors, the indicator must simultaneously contrast with both background colors. Additionally, the indicator color is often constrained to be chosen from a particular range or set (palette) of colors.

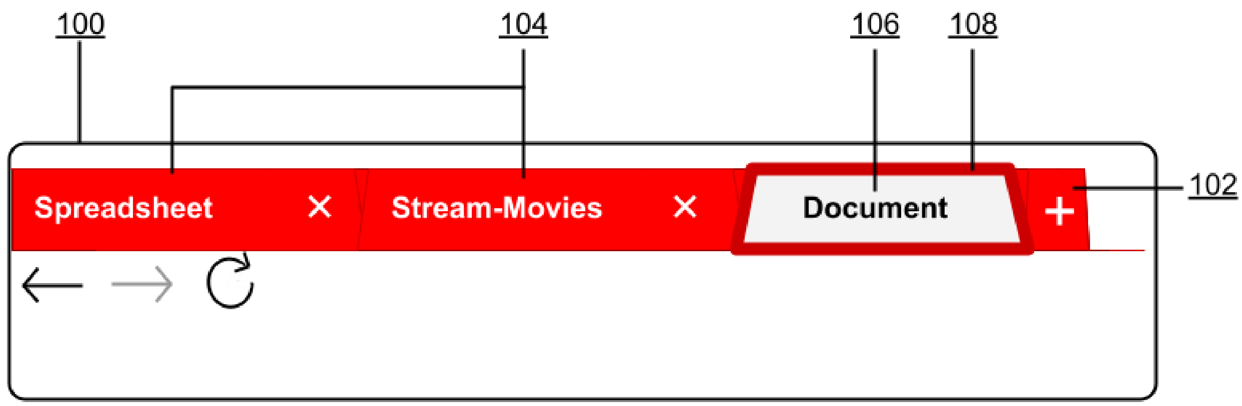

Fig. 1: An example of a foreground color failing to contrast with two background colors

Fig. 1 illustrates an example of a foreground (or state indicator) color that fails to contrast with two background colors. A browser (100) has a tabstrip (102). Tabs are displayed atop a background window-frame area. Inactive tabs are displayed in a reddish color (104) and active tabs in a grayish color (106). For accessibility, a focus ring (108), a type of state indicator, is displayed around the active tab. To contrast with the gray of the active tab, the focus ring is rendered in magenta, which, however, doesn't contrast sufficiently with the window frame.

This example demonstrates that for a UI element or other state indicator (e.g., focus ring) that is adjacent to two background elements (active tab and window frame) to be easily visible, it optimally contrasts with both background elements. Furthermore, such UI elements (focus rings, active tabs, window frames) have standard colors across a software product such that the chosen contrasting colors are optimally as similar as possible to those standard colors.

WCAG-2.1 (web content accessibility guidelines) describes techniques to measure the relative luminance and contrast between colors and sets various minimum contrast ratios for accessibility [1]. However, WCAG-2.1 does not describe techniques for automatically determining a contrasting color, let alone identifying one that contrasts with two distinct colors.

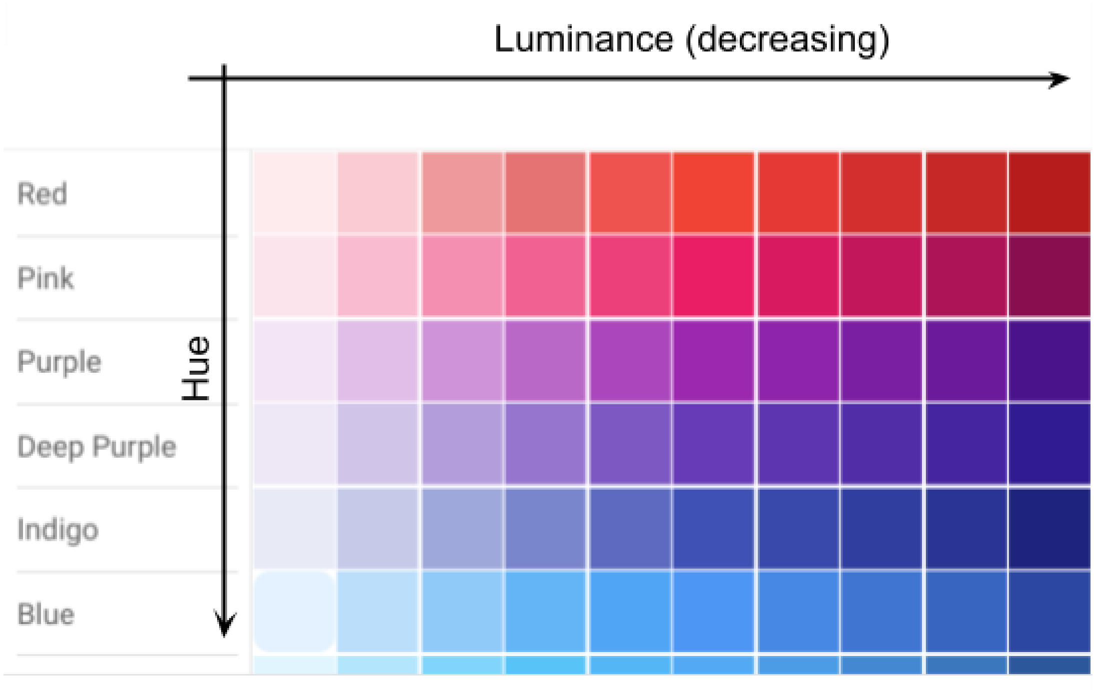

Fig. 2: A typical color palette includes colors indexed by hue and luminance

Fig. 2 illustrates a typical color palette. Colors are indexed by their hue along the Y-axis and, along the X-axis, by their luminance, or how light or dark each color is.

DESCRIPTION

Given two background colors A and B and an initial (or source) foreground color F_S from a given color space (e.g., the sRGB space), this disclosure describes techniques to automatically select a final (or target) foreground color F_T that contrasts with both A and B and is similar in hue to the initial foreground color. The choice of colors can be constrained to a palette of valid colors grouped by similar hue such that the final foreground color is from a predefined input palette. A minimum contrast threshold is provided as input. For example, per WCAG-2.1, a minimum contrast ratio can be 3:1. The three colors A, B, and F_S are typically specified by a user interface (UI) designer of an app or other software product.

The contrast between colors depends not on their hues or saturations but on their luminances. Given a range of potential relative luminances and two points within that range, the most-contrasting luminance to both points is either one of the endpoints of the range or the point between the two luminances that contrasts equally with both. Per techniques described herein, these points are computed efficiently and used to test the contrast of as few colors as possible from the palette. A target color is identified from the specified palette that is close to the hue of source color while meeting or exceeding the specified contrast threshold with each of the background colors.

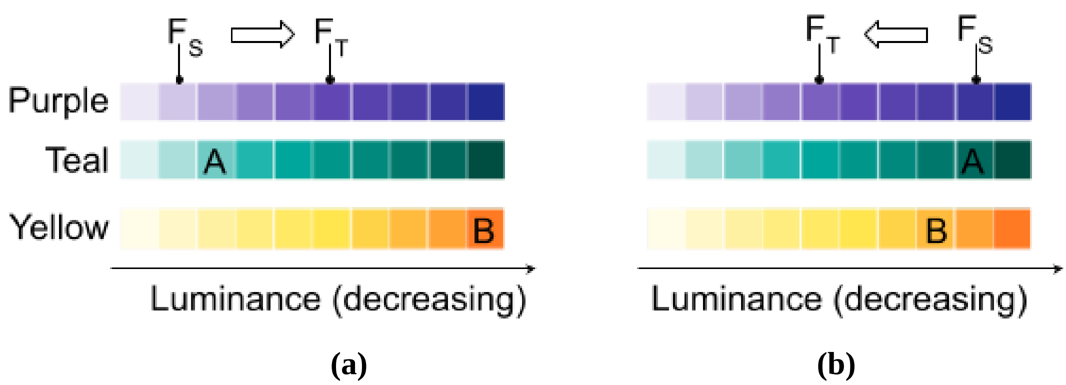

Fig. 3: (a) Computing a target foreground color F_T from a source foreground color F_S when the background colors A and B are: (a) near opposite endpoints of the luminance scale; and (b) near the same endpoint of the luminance scale

For example, in Fig. 3(a), the background colors A and B are near opposite ends of the luminance scale, with the source foreground color F_S being close in luminance to color A. For the foreground color to have an adequate contrast (e.g., luminance distance that exceeds the specified contrast threshold) to both background colors, the source color F_S is moved rightward to the middle of the luminance scale to arrive at the target color F_T. Effectively, the target color F_T is chosen such that it maximizes the smaller of the A-to-F_T and the B-to-F_T luminance distances.

In Fig. 3(b), the background colors A and B are near the same end of the luminance scale, with the source foreground color F_S also being at the same end. For the foreground color to have an adequate contrast to both background colors, the source color F_S is moved leftward. Although the target color can be placed at the extreme left of the luminance scale to provide a very high contrast with the background colors, it is adequate and advantageous to move it leftward just enough to provide a contrast that just exceeds the specified contrast threshold.

Given two background colors A and B and an initial (or source) foreground color F_S from a given color space (e.g., the sRGB space), a palette of valid colors grouped by hue, and a minimum contrast threshold, the final (target) foreground color F_T is computed as follows:

1. Precompute a representative hue for each similar color range in the palette.
2. Transform F_S to its hue-saturation-lightness (HSL) representation [3].
3. Select the palette color range whose representative hue is numerically closest to the computed hue (H-value) of F_S.
4. Compute the relative luminances of A, B, F_S, and the colors in the selected palette range [4].
5. Find the largest (lum_darkest) and the smallest (lum_lightest) relative luminances in the output range.
6. Compute the relative luminance midpoint (lum_range_midpoint) of the output range as the geometric mean of the darkest and the lightest relative luminances, e.g.,

```
lum_range_midpoint =
sqrt( (lum_lightest + 0.05) * (lum_darkest + 0.05) ) – 0.05.
```

7. Select as an initial output color the output color (from the selected range) with the closest relative luminance to that of F_S. Here and below, "closest relative luminance" means "contrasts least with", not "has the smallest numerical distance from".

8. Compute a target luminance lum_target using, for example, the pseudocode of Fig. 4.

```
auto contrast_ratio = [](float lum_1, float lum_2) {
  lum_1 += 0.05;
  lum_2 += 0.05;
  return (lum_1 > lum_2) ? (lum_1 / lum_2) : (lum_2 / lum_1);
};

lum_a = /* Relative luminance of A computed in step 4 */;
lum_b = /* Relative luminance of B computed in step 4 */;
lum_lightest = /* Lightest luminance computed in step 5 */;
lum_darkest = /* Darkest luminance computed in step 5 */;
lum_range_midpoint = /* Midpoint luminance computed in step 6 */;

lum_midpoint = sqrt((lum_a + 0.05) * (lum_b + 0.05)) - 0.05;
lum_endpoint = (lum_midpoint < lum_range_midpoint) ? lum_lightest : lum_darkest;

lum_lower_contrast = ((lum_a > lum_b) == (lum_endpoint > lum_midpoint)) ? lum_a :
lum_b;

con_midpoint = contrast_ratio(lum_lower_contrast, lum_midpoint);
con_endpoint = contrast_ratio(lum_lower_contrast, lum_endpoint);

lum_target = (con_midpoint > con_endpoint) ? lum_midpoint : lum_endpoint;
```

Fig. 4: Pseudocode to compute a target illuminance

As explained earlier, the example pseudocode of Fig. 4 computes `lum_target` as the luminance that contrasts most with both background colors, e.g., which maximizes the smaller of the `(lum_a, lum_target)` and the `(lum_b, lum_target)` distances.

9. Select as a target output color F_T the output color with the closest relative luminance to the target value from step 8.

10. Step through the output colors from the initial color (step 7) to the target (step 9) by checking the contrast between relative luminance of the selected color and by testing for the condition

`(((lum_a > lum_b) == (lum_selected > lum_midpoint)) ? lum_a : lum_b),`

where `lum_selected` is the relative luminance of the selected color.

11. Once the contrast exceeds the minimum threshold or the selected color is the target, return the selected color.

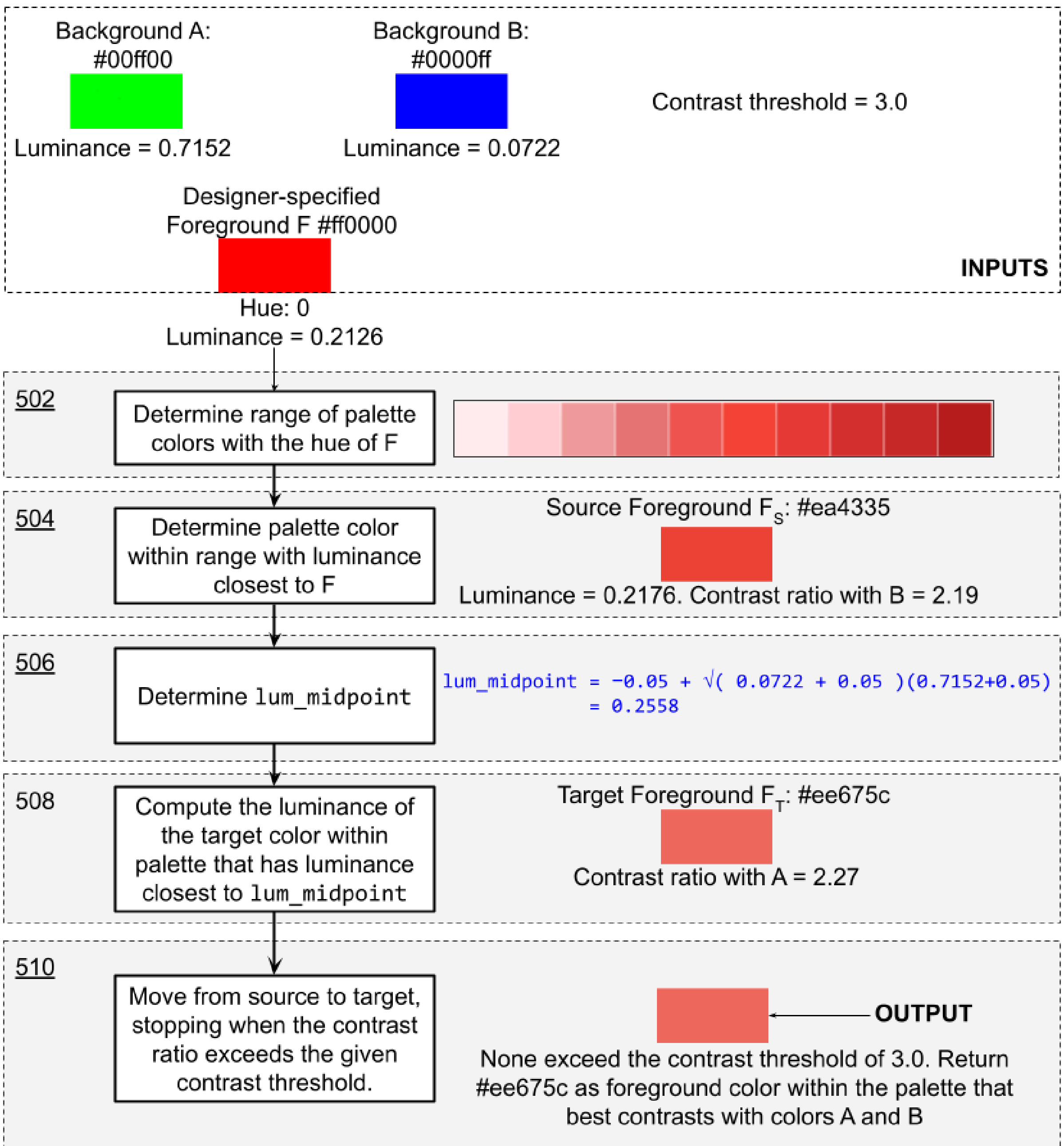

Fig. 5: A worked-out example

Fig. 5 illustrates a worked-out example. A designer specifies that the background color A be pure green (#00ff00), the background color B be pure blue (#0000ff), and the foreground color be red-like, e.g., pure red (#ff0000). The contrast threshold is specified as 3.0. A palette is specified such that the foreground color that optimizes the contrast between both background colors is to be picked from the palette. The particular pure red color (with hue-value = 0 and luminance 0.2126) specified for the foreground does not exist in the palette. Therefore, a range of palette colors is found with hue close to the hue specified by the designer for the foreground (502). The palette color within that range with luminance closest to the luminance of the specified foreground color is found (504) to be #ea4335, with a luminance 0.2176.

White does not contrast well against background A, and gray does not against background B, so the target luminance is the midpoint of the luminances of A and B. Since the luminances of the background colors are at different ends of the luminance scale, their midpoint is determined (506) as the geometric mean (0.2558) of their luminances. The target color, which is that color from the range with luminance closest to the midpoint, is determined to be a shade of red (508) with index #ee675c. This #ee675c shade of red has a contrast ratio of 2.27 with background color A, below the specified minimum contrast threshold of 3.0. To reach the specified threshold, colors are tested between source and target (510), but none are found. Therefore, the red color #ee675c is returned as that palette color that contrasts most with the given background colors pure green (#00ff00) and pure blue (#0000ff).

Compared to techniques that use a single indicator color or techniques that adjust the indicator to contrast against a single nearby background color, the described techniques improve user interface accessibility and usability. Compared to two-color indicator techniques, such as a white ring outside a black ring, the described techniques can provide a more aesthetically appealing user interface. A foreground color that sufficiently contrasts with two background colors is identified, improving legibility while maintaining the look-and-feel of the product.

CONCLUSION

Given two background colors and an initial foreground color, this disclosure describes techniques to automatically select a foreground color that contrasts with both background colors and is similar in hue to the initial foreground color. The choice of colors can be constrained to a palette of valid colors grouped by similar hue. Given a range of potential relative luminances and two points within that range, the most contrasting luminance to both points is determined to be an endpoint of the range or the midpoint between the two luminances.

REFERENCES

[1] “Web Content Accessibility Guidelines (WCAG) 2.1” available online at https://www.w3.org/TR/WCAG/#non-text-contrast accessed May 31, 2022.

[2] “Issue 1278449: Focus Contrast Is Too Low in the Tab Bar with both Windows 11 Accent Colors in the Title Bar+Window Borders and Pervasive System Accent Color” available online at https://bugs.chromium.org/p/chromium/issues/detail?id=1278449 accessed May 31, 2022.

[3] “RGB to HSL converter | color conversion” available online at https://www.rapidtables.com/convert/color/rgb-to-hsl.html accessed May 31, 2022.

[4] “Web Content Accessibility Guidelines (WCAG) 2.0” available online at http://www.w3.org/TR/WCAG20/#relativeluminancedef accessed May 31, 2022.