

Deriving New Oklab Exponents

To do

Original prompt

Here are 3 gamma values for 3 surrounds -

White Surround: Uses one exponent (e.g., 2.3). -

Gray Surround: Uses a slightly different value (e.g., 3.0). -

Dark Surround: Uses yet another exponent (e.g., 3.6).

Given these gamma values, could simple aoklab functions be generated that work for only a particular exponent? for example aoklab-white would work be reoptimized for an exponent 2.3. In this way, the functions could be as simple as the original?

Then original derivation prompt

The exponents 2.3, 3, and 3.6 were starting points that work for cielab L*. oklab L is different so we need to adjust the exponents.

Here is the mapping. for example, sRGB values that result in oklab lightness of 67, when the new white exponent is used, will result in lightness-adapted of 50.

oklab (1/3) 53.4 maps to oklab (dark) of 50

oklab (1/3) 59.8 maps to oklab (gray) of 50

oklab (1/3) 67 maps to oklab (white) of 50

gray L of 50 changes to L-adapted of 59.8

white L of 50 changes to L-adapted of 67

Adapted targets v2

- oklab (1/3) 41.7 maps to oklab (dark) of 40
- oklab (/13) 55.9 maps to oklab (white) of 40
- oklab (1/3) 48.3 maps to oklab (gray) of 40

first derivation

Here's the reformatted text for display in Obsidian, with proper math formatting:

Oklab Lightness Adaptation

The exponents 2.3, 3, and 3.6 were starting points that work for CIELAB L^* . Oklab L is different so we need to adjust the exponents.

Here is the mapping. For example, sRGB values that result in Oklab lightness of 67, when the new white exponent is used, will result in lightness-adapted of 50.

$$\text{Oklab}_{(1/3)} : 53.4 \mapsto \text{Oklab}_{(dark)} : 50$$

$$\text{Oklab}_{(1/3)} : 59.8 \mapsto \text{Oklab}_{(gray)} : 50$$

$$\text{Oklab}_{(1/3)} : 67 \mapsto \text{Oklab}_{(white)} : 50$$

Gray $L : 50$ changes to L -adapted 59.8

White $L : 50$ changes to L -adapted 67

Derivation

We can derive the "new exponents" by thinking in terms of how Oklab L is computed. In the original Oklab, a power-function with exponent $1/3$ is used. In a simplified picture, if we assume that (for a given channel or "representative" value) the Oklab lightness is roughly proportional to

$$L = 100, (x)^{1/3}$$

then for a given input x we have

$$x = \left(\frac{L}{100} \right)^3$$

Now suppose we wish to "remap" the lightness so that a value computed as L_{old} using exponent $1/3$ will instead yield a new lightness L_{new} when using a new exponent p . For a given representative x we have:

- Using the standard exponent: $x^{1/3} = \frac{L_{\text{old}}}{100}$
- With the new exponent p : $x^p = \frac{L_{\text{new}}}{100}$

Taking logarithms of the second equation and eliminating x (by writing $x = (L_{\text{old}}/100)^3$) gives

$$(L_{\text{old}}/100)^3 \text{ raised to } p \implies \left(\frac{L_{\text{old}}}{100} \right)^{3p} = \frac{L_{\text{new}}}{100}$$

Solving for p we get:

$$3p = \frac{\ln(L_{\text{new}}/100)}{\ln(L_{\text{old}}/100)}$$

$$p = \frac{1}{3}, \frac{\ln(L_{\text{new}}/100)}{\ln(L_{\text{old}}/100)}$$

Using the mappings provided, let's compute p (the new exponent) for each surround. (We then note that "gamma" is the reciprocal of the exponent.)

1. White Surround

For white, you stated that an sRGB that produces an original Oklab L of 67 should instead give an adapted L of 50. That is:

- $L_{\text{old}} = 67$ and $L_{\text{new}} = 50$

Compute:

$$p_{\text{white}} = \frac{1}{3}, \frac{\ln(50/100)}{\ln(67/100)} = \frac{1}{3}, \frac{\ln(0.50)}{\ln(0.67)}$$

Numerically:

- $\ln(0.50) \approx -0.693147$
- $\ln(0.67) \approx -0.400477$

Thus,

$$p_{\text{white}} \approx \frac{1}{3}, \frac{-0.693147}{-0.400477} \approx \frac{1}{3}, (1.730) \approx 0.5767$$

The corresponding gamma (reciprocal of the exponent) is about

$$\gamma_{\text{white}} \approx \frac{1}{0.5767} \approx 1.734$$

2. Gray Surround

For gray, you provided that an original L of 59.8 should map to 50.

- $L_{\text{old}} = 59.8$ and $L_{\text{new}} = 50$

Then:

$$p_{\text{gray}} = \frac{1}{3}, \frac{\ln(0.50)}{\ln(59.8/100)}$$

First, $\ln(59.8/100) = \ln(0.598) \approx -0.5148$. Thus:

$$p_{\text{gray}} \approx \frac{1}{3}, \frac{-0.693147}{-0.5148} \approx \frac{1}{3}, (1.346) \approx 0.4487$$

And the new gamma is:

$$\gamma_{\text{gray}} \approx \frac{1}{0.4487} \approx 2.23$$

3. Dark Surround

For dark, an original L of 53.4 should map to 50.

- $L_{\text{old}} = 53.4$ and $L_{\text{new}} = 50$

Compute:

$$p_{\text{dark}} = \frac{1}{3}, \frac{\ln(0.50)}{\ln(53.4/100)}$$

Now, $\ln(53.4/100) = \ln(0.534) \approx -0.6276$. So:

$$p_{\text{dark}} \approx \frac{1}{3}, \frac{-0.693147}{-0.6276} \approx \frac{1}{3}, (1.1046) \approx 0.3682$$

And the corresponding gamma:

$$\gamma_{\text{dark}} \approx \frac{1}{0.3682} \approx 2.72$$

Summary of New Exponents and Gammas

- **White Surround:**
New exponent $p_{\text{white}} \approx 0.577$ (i.e. gamma ≈ 1.73)
- **Gray Surround:**
New exponent $p_{\text{gray}} \approx 0.449$ (i.e. gamma ≈ 2.23)
- **Dark Surround:**
New exponent $p_{\text{dark}} \approx 0.368$ (i.e. gamma ≈ 2.72)

These new exponents (when applied uniformly in the nonlinearity step) should yield adapted Oklab lightness values of 50 for the representative cases, thus "compressing" the original lightness (67, 59.8, 53.4) to a uniform 50. In a full implementation, these exponents would be "baked into" the conversion functions and their associated (reoptimized) matrices to ensure that hue uniformity is preserved while the lightness is remapped as desired.

Failed reoptimization

I've tested the improved adaptive oklab library. it works well and the exponents are close. They need to be adjusted. colors using standard

oklab with a 1/3 exponent leading to a lightness of 40 should map to

White: adaptedL = 55.9

Gray: adapted L = 48.3

Dark: adapted L = 41.7

It is interesting that standard oklab is nearly dark adapted. reoptimize the adaptiveoklab.js code and exponents for these new targets.

! Mistake is these should be mapped to 40, not 50

AdaptiveOklab.js Library Update

Below is the updated AdaptiveOklab.js library with the new target mappings. In this version, we want colors that would normally have a standard Oklab lightness of 40 (using the 1/3 exponent) to be remapped as follows:

- **White Surround:** Adapted L = 55.9
- **Gray Surround:** Adapted L = 48.3
- **Dark Surround:** Adapted L = 41.7

Because in standard Oklab lightness is computed approximately as:

$$L_{\text{std}} = 100, x^{1/3}$$

an L of 40 implies:

$$x = \left(\frac{40}{100} \right)^3 = 0.064$$

For an adaptive exponent p we want:

$$L_{\text{adapt}} = 100, x^p$$

to equal the target value. For each surround, we solve:

$$100, (0.064)^p = L_{\text{target}} \implies (0.064)^p = \frac{L_{\text{target}}}{100}$$

Taking natural logs gives:

$$p = \frac{\ln(L_{\text{target}}/100)}{\ln(0.064)}$$

Using this formula we get:

- **White Surround:** $L_{\text{target}} = 55.9$ so

$$p_{\text{white}} = \frac{\ln(0.559)}{\ln(0.064)} \approx \frac{-0.581}{-2.748} \approx 0.2115$$

- **Gray Surround:** $L_{\text{target}} = 48.3$ so

$$p_{\text{gray}} = \frac{\ln(0.483)}{\ln(0.064)} \approx \frac{-0.727}{-2.748} \approx 0.2648$$

- **Dark Surround:** $L_{\text{target}} = 41.7$ so

$$p_{\text{dark}} = \frac{\ln(0.417)}{\ln(0.064)} \approx \frac{-0.874}{-2.748} \approx 0.3180$$

Notice that standard Oklab uses an exponent of $1/3 \approx 0.3333$, so the dark condition (with $p \approx 0.3180$) is very similar to the standard—consistent with the observation that standard Oklab is nearly dark adapted.

To preserve hue uniformity we continue to apply the same power-law nonlinearity uniformly across all LMS channels. Because the matrices used in Oklab were optimized for an exponent of $1/3$, we compensate for the change by applying a correction factor to the opponent channels (a and b):

$$\text{correction} = x_0^{(\frac{1}{3}-p)}$$

with x_0 a representative LMS value (here taken as 0.5).

Corrected derivation

Let's re-derive the mapping carefully. In standard Oklab the lightness is computed as

$$L_{\text{std}} = 100 \cdot x^{(1/3)},$$

so that

$$x = (L_{\text{std}}/100)^3.$$

In the adaptive system we want to use a new exponent p such that

$$L_{\text{adapt}} = 100 \cdot x^{(p)}.$$

For a given color with standard Oklab lightness L_{std} , we then have

$$L_{\text{adapt}} = 100 \cdot ((L_{\text{std}}/100)^3)^{(p)} = 100 \cdot (L_{\text{std}}/100)^{(3p)}.$$

We want this adaptive lightness to be 40. That is, for each condition:

$$100 \cdot (L_{\text{std}}/100)^{(3p)} = 40.$$

Dividing by 100 gives

$$(L_{\text{std}}/100)^{(3p)} = 0.40.$$

Taking natural logs yields

$$3p \cdot \ln(L_{\text{std}}/100) = \ln(0.40),$$

so that

$$p = \ln(0.40) / [3 \cdot \ln(L_{\text{std}}/100)].$$

Now plug in the given standard Oklab lightness values for each surround:

Dark Surround

Standard Oklab value: $L_{\text{std}} = 41.7$

1. Compute $\ln(41.7/100) = \ln(0.417)$.
Approximately, $\ln(0.417) \approx -0.874$.
2. $\ln(0.40) \approx -0.9163$.

Thus, $p_{\text{dark}} = (-0.9163) / [3 \cdot (-0.874)] \approx 0.9163 / 2.622 \approx 0.349$.

Gray Surround

Standard Oklab value: $L_{\text{std}} = 48.3$

1. Compute $\ln(48.3/100) = \ln(0.483)$.
Approximately, $\ln(0.483) \approx -0.727$.
2. $\ln(0.40) \approx -0.9163$.

Thus, $p_{\text{gray}} = (-0.9163) / [3 \cdot (-0.727)] \approx 0.9163 / 2.181 \approx 0.420$.

White Surround

Standard Oklab value: $L_{\text{std}} = 55.9$

1. Compute $\ln(55.9/100) = \ln(0.559)$.
Approximately, $\ln(0.559) \approx -0.581$.

2. $\ln(0.40) \approx -0.9163$.

Thus, $p_{\text{white}} = (-0.9163) / [3 \cdot (-0.581)] \approx 0.9163 / 1.743 \approx 0.526$.

Summary of New Exponents

- **White Surround:** $p_{\text{white}} \approx 0.526$
- **Gray Surround:** $p_{\text{gray}} \approx 0.420$
- **Dark Surround:** $p_{\text{dark}} \approx 0.349$

These values mean that if a color computed with the standard Oklab (using exponent $1/3 \approx 0.333$) yields $L_{\text{std}} = 55.9$ (white), 48.3 (gray), or 41.7 (dark), then applying the adaptive conversion with these exponents will remap that lightness to 40.

Notice that compared to the standard exponent of 0.333, all three new exponents are higher—since x (computed from L_{std}) is less than 1, raising it to a higher exponent yields a lower number. That’s why the adapted lightness comes out lower (40) than the standard lightness.

These new exponents (0.526 for white, 0.420 for gray, and 0.349 for dark) are relatively close together and represent only small adjustments away from the standard Oklab value. They now serve as the “baked-in” exponents for the adaptive conversion for each surround condition.

Final Answer

The correct adaptive exponents for mapping standard Oklab lightness values to 40 are:

- **White Surround:** $p \approx 0.526$
- **Gray Surround:** $p \approx 0.420$
- **Dark Surround:** $p \approx 0.349$

These are derived from the equation

$$p = \ln(0.40) / [3 \cdot \ln(L_{\text{std}}/100)],$$

using $L_{\text{std}} = 55.9$ (white), 48.3 (gray), and 41.7 (dark), respectively.

Below is the regenerated adaptiveoklab.js library using the new baked-in exponents. In this version, the conversion is designed so that a color whose standard Oklab lightness (computed with the $1/3$ exponent) is:

- 55.9 (white),
- 48.3 (gray), or
- 41.7 (dark)

will be remapped to an adapted lightness of 40. These exponents were derived from the relation

$$p = \ln(0.40) / [3 \cdot \ln(L_{\text{std}}/100)]$$

yielding:

- White: $p \approx 0.526$
- Gray: $p \approx 0.420$
- Dark: $p \approx 0.349$

A correction factor is applied to the opponent channels (a and b) to preserve hue uniformity relative to the standard Oklab (with exponent $1/3 \approx 0.3333$).

```
// adaptiveoklab.js

/**
 * AdaptiveOklab.js
 *
 * This library implements an adaptive version of Oklab conversion that
 * remaps
 * lightness based on the viewing surround while preserving hue uniformity.
 *
 * In standard Oklab, lightness is computed approximately as:
 *   L_std = 100 · x^(1/3),
 * where x is the nonlinearly compressed cone response.
 *
 * For a given standard Oklab lightness L_std, we have:
 *   x = (L_std/100)^3.
 *
 * We want to remap colors so that they all have an adapted lightness
 * L_adapt = 40.
 *
 * The target standard Oklab lightness values for different surrounds are:
 *   - White Surround: L_std = 55.9
 *   - Gray Surround:  L_std = 48.3
 *   - Dark Surround:  L_std = 41.7
 *
```

```

* For each case, we require:
*    $100 \cdot ((L\_std/100)^3)^p = 40.$ 
*
* Solving for the adaptive exponent p gives:
*    $p = \ln(0.40) / [3 \cdot \ln(L\_std/100)].$ 
*
* This yields:
*   - White:  $p \approx 0.526$ 
*   - Gray:   $p \approx 0.420$ 
*   - Dark:   $p \approx 0.349$ 
*
* To preserve hue, the same power function is applied to every LMS channel.
* Because the LMS-to-Oklab conversion matrix was optimized for the standard
exponent
* of 1/3, we apply a correction factor to the opponent channels (a and b):
*
*    $correction = x_0^{((1/3) - p)},$ 
*
* where  $x_0$  is a representative LMS value (here taken as 0.5).
*/

```

```

export class AdaptiveOklab {
  /**
   * Creates an instance of AdaptiveOklab.
   *
   * @param {Object} options - Configuration options.
   * @param {string} options.surround - One of "white", "gray", or "dark".
Default is "gray".
   * @param {number} [options.x0=0.5] - Representative LMS value used for
computing the correction factor.
   */
  constructor(options = {}) {
    // Choose the surround condition.
    this.surround = options.surround || 'gray';

    // Baked-in exponents for each surround condition, derived from:
    //    $p = \ln(0.40) / [3 \cdot \ln(L\_std/100)]$ 
    // where L_std is the standard Oklab lightness value (using exponent
1/3) that should map to 40.
    //   - For white:  L_std = 55.9 →  $p \approx 0.526$ 
    //   - For gray:   L_std = 48.3 →  $p \approx 0.420$ 
    //   - For dark:   L_std = 41.7 →  $p \approx 0.349$ 
    this.surroundExponents = {
      white: 0.526,
      gray:  0.420,
      dark:  0.349
    }
  }
}

```

```

};

// Use the exponent for the chosen surround.
this.exponent = this.surroundExponents[this.surround] ||
this.surroundExponents.gray;

// Representative LMS value for computing the correction factor.
// Correction factor =  $x_0^{(1/3) - p}$ , with  $1/3 \approx 0.3333$  being the
standard Oklab exponent.
this.x0 = options.x0 || 0.5;

// Matrix for converting from XYZ to LMS. (Fixed.)
this.M_XYZ_to_LMS = [
  [ 0.8189330101, 0.3618667424, -0.1288597137],
  [ 0.0329845436, 0.9293118715, 0.0361456387],
  [ 0.0482003018, 0.2643662691, 0.6338517070]
];

// Matrix for converting from nonlinearly compressed LMS to Oklab.
// These coefficients were originally optimized for the standard
exponent (1/3).
this.M_LMS_to_Oklab = [
  [ 0.2104542553, 0.7936177850, -0.0040720468],
  [ 1.9779984951, -2.4285922050, 0.4505937099],
  [ 0.0259040371, 0.7827717662, -0.8086757660]
];
}

/**
 * Applies a sign-preserving power function.
 *
 * @param {number} value - The input value.
 * @param {number} exponent - The exponent to use.
 * @returns {number} The value after applying the power function.
 */
signPreservingPow(value, exponent) {
  return Math.sign(value) * Math.pow(Math.abs(value), exponent);
}

/**
 * Multiplies a 3x3 matrix by a 3x1 vector.
 *
 * @param {Array<Array<number>>}> matrix - The 3x3 matrix.
 * @param {Array<number>} vector - The 3-element vector.
 * @returns {Array<number>} The resulting 3-element vector.
 */

```

```

multiplyMatrixVector(matrix, vector) {
  return matrix.map(row =>
    row.reduce((acc, cur, i) => acc + cur * vector[i], 0)
  );
}

/**
 * Computes the correction factor for the opponent channels (a and b)
 * to preserve hue uniformity.
 *
 * The correction factor is defined as:
 *   correction = x0(1/3) - p,
 * where 1/3 (≈0.3333) is the standard Oklab exponent and p is the new
 * exponent.
 *
 * @returns {number} The correction factor.
 */
getCorrectionFactor() {
  const standardExponent = 1 / 3.0; // ≈0.3333
  return Math.pow(this.x0, standardExponent - this.exponent);
}

/**
 * Converts from XYZ to Adaptive Oklab.
 *
 * @param {number} X - The X component of XYZ.
 * @param {number} Y - The Y component of XYZ.
 * @param {number} Z - The Z component of XYZ.
 * @returns {Object} An object containing L, a, and b.
 */
fromXYZ(X, Y, Z) {
  // 1. Convert from XYZ to LMS.
  const LMS = this.multiplyMatrixVector(this.M_XYZ_to_LMS, [X, Y, Z]);

  // 2. Apply the adaptive power-law nonlinearity using the baked-in
  // exponent.
  const LMSprime = LMS.map(channel => this.signPreservingPow(channel,
    this.exponent));

  // 3. Convert from the nonlinearly compressed LMS to Oklab.
  let lab = this.multiplyMatrixVector(this.M_LMS_to_Oklab, LMSprime);

  // 4. Apply the correction factor to the opponent channels (a and b)
  // to preserve hue uniformity.
  const correction = this.getCorrectionFactor();
  lab[1] *= correction;
}

```

```

    lab[2] *= correction;

    return { L: lab[0], a: lab[1], b: lab[2] };
}

/**
 * Converts from sRGB to Adaptive Oklab.
 * Assumes sRGB values are in the range [0, 1].
 *
 * @param {number} r - The red channel.
 * @param {number} g - The green channel.
 * @param {number} b - The blue channel.
 * @returns {Object} An object containing L, a, and b.
 */
fromSRGB(r, g, b) {
    // Convert sRGB to linear sRGB.
    const linearize = (v) =>
        v <= 0.04045 ? v / 12.92 : Math.pow((v + 0.055) / 1.055, 2.4);
    const rLin = linearize(r);
    const gLin = linearize(g);
    const bLin = linearize(b);

    // Convert linear sRGB to XYZ using the standard transformation matrix.
    const X = 0.4124564 * rLin + 0.3575761 * gLin + 0.1804375 * bLin;
    const Y = 0.2126729 * rLin + 0.7151522 * gLin + 0.0721750 * bLin;
    const Z = 0.0193339 * rLin + 0.1191920 * gLin + 0.9503041 * bLin;

    return this.fromXYZ(X, Y, Z);
}
}

```

Usage Example

```

import { AdaptiveOklab } from './adaptiveoklab.js';

// Create an AdaptiveOklab instance for a "white" surround.
const aok = new AdaptiveOklab({ surround: 'white' });

// Convert an sRGB color (values in [0, 1]) to Adaptive Oklab.
const lab = aok.fromSRGB(0.2, 0.5, 0.8);
console.log(`L: ${lab.L}, a: ${lab.a}, b: ${lab.b}`);

```

Summary

This regenerated library uses the following baked-in exponents:

- **White Surround:** $p \approx 0.526$
- **Gray Surround:** $p \approx 0.420$
- **Dark Surround:** $p \approx 0.349$

These values were derived so that standard Oklab lightness values of 55.9, 48.3, and 41.7 (computed with a $1/3$ exponent) are remapped to an adapted lightness of 40, while preserving hue uniformity through a correction factor applied to the opponent channels.