CHROMATIC CONTINUITY FRAMEWORK: PERCEPTUAL COLOR FOR DESIGN SYSTEMS IN THE GENERATIVE ERA

Hrozian S., *Staff Product Designer, Speak, San Francisco, California, USA*

Abstract. *This thesis introduces the Chromatic Continuity Framework (CCF) – a four-principle approach for designing color systems that survive AI-driven generation. Drawing on perceptual color science (OKLCH), production design system practice, and the APCA accessibility standard, CCF argues that color must be defined in human-meaningful coordinates before generation, not validated after. Examples from Sunmile, iOS, Material Design, and Tailwind illustrate the framework.*

Keywords: *color management, OKLCH, design systems, generative AI, accessibility, perceptual color.*

Color in digital design has been hex or RGB for thirty years. Both formats were engineered for screens. Not for human perception. Generative AI now produces full color systems on demand [1, 2]. The legacy convention is a liability. Tools that "know" color through hex generate palettes that drift across light and dark modes, fail accessibility checks, and lose semantic intent. Browsers caught up first. CSS Color Module Level 4 [3] shipped OKLCH support. Design tooling is catching up. Generative tools still receive hex as input. The gap is not whether AI can produce color. Every model already does. The gap is whether the inputs we give AI describe color in a way that survives generation. Continuity is the design problem.

This thesis introduces the Chromatic Continuity Framework (CCF). Four principles for color systems built to remain coherent under AI-driven generation. The framework names what most production color systems already do partially and ties them into one stack (Table 1).

Table 1 – Four principles for AI-aware color systems

Principle	**Definition**
Perceptual Anchoring	Define color in perceptual space (OKLCH) before generation, so AI works from human-meaningful coordinates rather than RGB.
Tonal Symmetry	Light and dark modes share lightness curves and chroma envelopes, not just inverted values, ensuring coherent variants.
Semantic Layering	Tokens carry meaning (warning, success, brand) separately from raw color, letting AI swap palettes without breaking system intent.
Accessibility-First Generation	Contrast ratios are constraints input before generation, not validated after.

Perceptual Anchoring defines color in OKLCH before any generation step. The reason is operational. Generative tools extrapolate. RGB extrapolation produces visually inconsistent results because RGB distance does not match perceived distance [4]. Sunmile's tokens are authored in OKLCH and converted to RGB only at render time. Apple's P3 system tokens follow the same principle [5]. Tonal Symmetry requires that light and dark modes share lightness curves and chroma envelopes. Not inverted values. Tailwind v4's OKLCH palette generates colors from a single hue and chroma rule. Dark-mode variants preserve perceived contrast as a result (Figure 1).

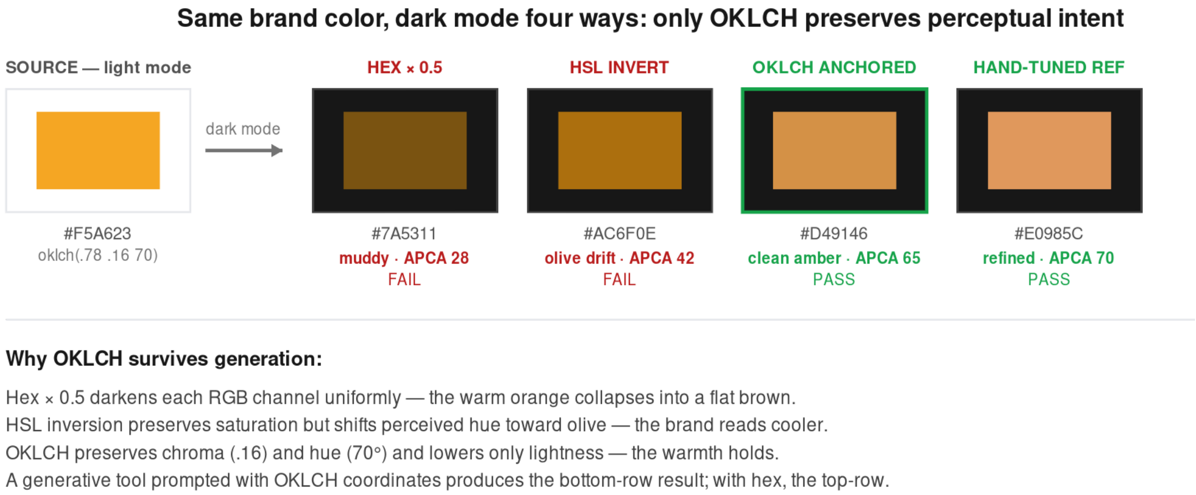

Figure 1 – The same brand color taken to dark mode four ways.
Only OKLCH preserves the warmth, the chroma, and APCA-passing contrast

Semantic Layering keeps meaning separate from raw color. A token named feedback.warning carries the intent of a warning. The OKLCH coordinates are an implementation detail that can change per theme without breaking system intent. Accessibility-First Generation treats contrast as a constraint passed into generation. Not a check applied after. APCA [6] provides a measurement that survives perceptual reasoning. Older WCAG ratios do not. Sunmile uses APCA in token authoring. Generated themes that pass the APCA constraint pass for the same perceptual reason a designer would.

CCF imposes upfront cost. Translating a hex-based system into perceptual coordinates takes weeks of careful work, and most production pipelines still lack OKLCH support end to end. Designers trained on hex have to learn perceptual reasoning, which is genuinely a new skill [7]. Honestly, the framework is most useful for new systems. Retrofitting an existing one is harder than it looks. Provenance is a related design problem here. A theme produced by an AI tool should carry that fact in its token metadata, not be passed off as hand-authored brand color.

AI tools can produce color now. That is the settled part. The harder question is whether the color stays coherent across light and dark, across accessibility thresholds, across the kind of perceptual distance only experienced designers tend to notice. Hex was the wrong abstraction for this, honestly. OKLCH is closer to how the eye actually works, which is the test that matters in the end.

References

1. Kaluhin, N., Vovk, O., & Chebotarova, I. (2024). The impact of artificial intelligence on future of humanity. Jóvenes en la ciencia, (26). https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/4235/3716.

2. Chebotarova, I., & Cherkashyna, H. (2025). Font management with artificial intelligence. Memorias de SYNTOPIA. (p. 32-33).

3. Lilley, C., Verou, L., & Atanassov, A. (2022). CSS Color Module Level 4. W3C Candidate Recommendation. https://www.w3.org/TR/css-color-4/.

4. Ottosson, B. (2020). A perceptual color space for image processing. https://bottosson.github.io/posts/oklab/.

5. Apple Inc. (2024). Human Interface Guidelines: Color. https://developer.apple.com/design/human-interface-guidelines/color.

6. Somers, A. (2023). APCA: Accessible Perceptual Contrast Algorithm. https://git.apcacontrast.com/.

7. Fairchild, M. D. (2013). Color appearance models (3rd ed.). Wiley.