A Generated Benchmark of Metameric Pairs for Testing Color Pipelines Against the Illuminant Assumption

Satyajeeth Suresh Kannan
Innovation Academy · Mary's Lab
github.com/JeetuSK0808/maryslab

Preprint · August 2026 · Mary's Lab technical report TR-08

ABSTRACT

Color pipelines frequently encode an implicit assumption: that a surface has a color. The assumption survives testing because test images are rendered under one illuminant, where it is indistinguishable from truth. We generate a benchmark that breaks it on demand. Each item is a pair of reflectances constructed to be colorimetrically identical under a match illuminant and maximally divergent under a split illuminant, with both quantities reported per item. Over 12 generated items the mean residual disagreement under the match illuminant is 0 ΔE_{OK} — numerically indistinguishable from identity — while the median divergence under the split illuminant is 0.037 and the maximum 0.0498. Any stage that treats color as intrinsic to a surface fails visibly on such a pair, and the failure is diagnostic rather than statistical.

Keywords: benchmark; metamerism; software testing; color management; reproducibility

1. Introduction

A test suite for a color pipeline typically renders known inputs and compares against known outputs. If every test renders under D65, no test distinguishes a pipeline that models reflectance from one that stores a color and reuses it under any light. Both pass.

Metamerism supplies the discriminating input. Two surfaces that match under one illuminant and differ under another cannot both be represented correctly by a single color value. A pipeline that carries spectra through will show them diverging under the second light; a pipeline that collapsed them to a color will show them identical, and that is a bug made visible.

Constructing such pairs by hand is laborious, which is presumably why such tests are rare. We generate them.

2. Method

For each anchor color, we construct the metamer set under the match illuminant — the polytope of reflectances producing that color — and search within it for pairs maximising divergence under the split illuminant. Because both members are drawn from the same metamer set by construction, their match-illuminant agreement is exact up to numerical residual, and that residual is reported per item rather than assumed.

Generation is seeded and deterministic: the same seed reproduces the same benchmark, which is a requirement for a benchmark to be citable. Items carry their match color, both spectra, the match residual, and the split divergence, so a consumer can sort by difficulty.

3. Evaluation

Inputs are synthetic by design. A benchmark of this kind is necessarily generated: the property it tests — exact colorimetric agreement under one light with maximal divergence under another — is a construction, not something one finds in a material library at will.

Table 1. Benchmark statistics over generated items, match illuminant D65, split illuminant A. The match residual measures how exactly the pairs agree where they should; the split divergence measures how strongly they disagree where they should.

quantity	value
anchors	6
items generated	12
mean match residual (ΔE_OK)	0
max match residual (ΔE_OK)	0
median split divergence	0.037
max split divergence	0.0498
min split divergence	0.0122
seeds exhausted	0

The match residual is the quantity that validates the construction. At 0 in the worst case, the pairs agree under the match illuminant far below any perceptual threshold — they are metamers in the operative sense, not merely similar colors. The split divergence spans a usable difficulty range, allowing a test suite to include both subtle and unmistakable cases.

The intended use is a regression test. A pipeline is fed both members of a pair, rendered under the split illuminant, and required to produce different outputs. A pipeline that returns identical outputs has collapsed the spectra to a color somewhere, and the item identifies that.

Reproducing these numbers

Clone `github.com/JeetuSK0808/maryslab` and run `node papers/build/experiments.mjs`. Every figure in this report is written by that script into `papers/build/results.json`; the instrument itself is exercised by `generate_metamer_benchmark`. The engines carry a 401-vector conformance suite shared by a TypeScript reference implementation, a C++20 core, and a WebAssembly build, so a reported number is a property of the specification rather than of one implementation.

4. Real-data evaluation: anchoring the benchmark on measured colors

A benchmark's items are necessarily constructed — the property being tested is a construction. What need not be invented is the *colour* each item is built around. The anchors above were chosen for convenience, sitting well inside the object-colour solid where metamer sets are comfortably large. Anchoring instead on the measured appearance of real materials asks whether the generator still works where the colours actually are.

We drew 8 materials from `usgs-splib07a-1`, took each one's measured colour under D65 as an anchor, and generated against the same D65/A split.

Table 2. Benchmark generated from the measured colours of 8 USGS materials, match D65, split A. The synthetic run of §3 is shown alongside.

quantity	measured anchors	synthetic anchors (§3)
anchors	8	6
items generated	12	12
anchors yielding nothing	2	0
max match residual (ΔE_OK)	0	0
median split divergence	0.019	0.037
max split divergence	0.056	0.0498

The construction holds: max match residual is 0, so the pairs agree under D65 exactly, as they must to be metamers at all. Divergence under A spans a usable range and the median, 0.019, is close to the synthetic run's.

The informative difference is the row that reports nothing: 2 of the 8 measured anchors produced no items, against 0 for the synthetic set. Real material colours are more saturated and sit closer to the boundary of the object-colour solid, and a metamer set that has collapsed toward a single member has no room for a pair. This is the same phenomenon TR-07 measures directly as envelope width, arriving here as a generation failure rate — and it is the practical reason a benchmark built on real colours needs more anchors than one built on convenient ones.

Reproducing these numbers

Clone `github.com/JeetuSK0808/maryslab` and run `node papers/build/experiments.mjs`. Every figure in this report is written by that script into `papers/build/results.json`; the instrument itself is exercised by `generate_metamer_benchmark`. The engines carry a 401-vector conformance suite shared by a TypeScript reference implementation, a C++20 core, and a WebAssembly build, so a reported number is a property of the specification rather than of one implementation.

5. Limitations

This section states what the result does not establish. It is written to be read before the abstract is quoted.

- **Real anchors are still generated items.** Anchoring on a measured colour makes the target real; the two reflectances constructed to meet there remain constructions, and need not correspond to manufacturable materials.
- **Generated, not measured.** The reflectances are constructed within a metamer polytope and need not correspond to manufacturable materials. The benchmark tests software, not pigments.
- **Difficulty is not calibrated to human perception.** Split divergence is reported in ΔE_{OK}; no psychophysical study establishes what divergence a viewer notices under what conditions.
- **Two illuminants only.** Each item is built around a single match/split pair. Real failures may involve ob-

server variation or three-illuminant scenarios that this construction does not cover.

- **A passing pipeline is not a correct pipeline.** Surviving this benchmark shows only that colors were not collapsed under these conditions.
- **Prior art was surveyed informally.** Metamer generation is established; a search found no released, seeded benchmark of metameric pairs intended as a software regression suite.

6. Acknowledgements and disclosure

The instruments described here were implemented in Mary's Lab, an open-source computational color science project. Software design, implementation, and the preparation of this report were carried out with substantial assistance from a large language model (Anthropic Claude), under the author's direction and review. All numerical results were produced by executing the released code; none were generated by a language model. The author is responsible for the claims made here.

References

1. Finlayson, G. D. and Morovic, P. (2005). Metamer sets. *Journal of the Optical Society of America A*, 22(5), 810–819.
2. Logvinenko, A. D., Funt, B., and Godau, C. (2014). Metamer mismatching. *IEEE Transactions on Image Processing*, 23(1), 34–43.
3. Wyszecki, G. and Stiles, W. S. (1982). *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd edition. Wiley.
4. CIE (2004). *Colorimetry, 3rd edition*. CIE 15:2004. Commission Internationale de l'Éclairage, Vienna.
5. Ottosson, B. (2020). *A perceptual color space for image processing (Oklab)*. bottosson.github.io.
6. Chen, T. Y. et al. (2018). Metamorphic testing: a review of challenges and opportunities. *ACM Computing Surveys*, 51(1), 1–27.
7. Schanda, J. (ed.) (2007). *Colorimetry: Understanding the CIE System*. Wiley-Interscience.
8. Foster, D. H., et al. (2006). Frequency of metamerism in natural scenes. *Journal of the Optical Society of America A*, 23(10), 2359–2372.

Source, engines, and the script that produced every number in this report: `github.com/JeetuSK0808/maryslab`. Results regenerate with `node papers/build/experiments.mjs`. Inputs: synthetic where §3 says so, and measured USGS splib07a spectra (snapshot `usgs-splib07a-1`) where a real-data section says so. Prepared with AI assistance (see Acknowledgements).